

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 1

CENTRO DE SOFTWARE LIBRE



**Módulo para la administración remota del servicio
telemático DHCP en la plataforma Nova-ARST**

Trabajo de diploma para optar por el título de Ingeniero en Ciencias
Informáticas

Autor

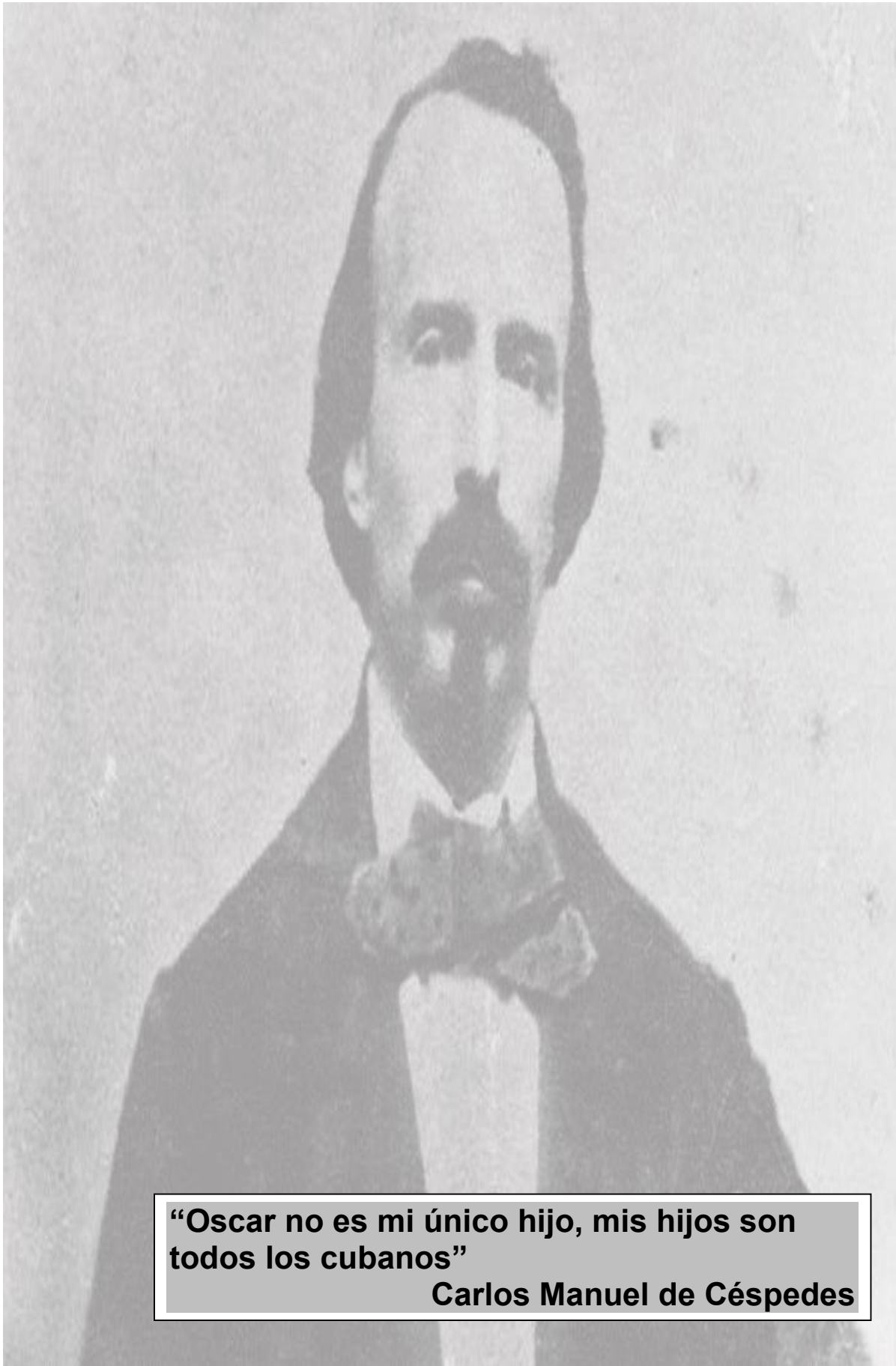
Reynier Fernández Rodríguez

Tutores

Ing. Profesor. Instructor Hanny Valdés Hernández

Ing. Enrique Muschett Cortina

La Habana, Noviembre del 2022



“Oscar no es mi único hijo, mis hijos son todos los cubanos”

Carlos Manuel de Céspedes

Agradecimientos

A través de este medio agradezco a todas aquellas personas que en la sombra hicieron posible este resultado maravilloso, en especial a mi madre querida la cual nunca me exigió llegar a este nivel, ni escoger mi carrera, solo dejó que tomara decisiones propias y las apoyó en todo momento, a ella mi más especial agradecimiento, gracias a May, quien me dio fuerzas para lograr esta meta, su apoyo incondicional fue un pilar para mi crecimiento como ser humano y profesional, a ella eternamente agradecida. Gracias aquellas personas que estuvieron en el pasado y ya no están cerca pero fueron parte de este proceso y a las que están ahora y con pequeñas acciones también han demostrado su apoyo. A mis tutores y cotutores, profesores, compañeros y hermanos que se hicieron en el camino, que la vida les devuelva a todos ustedes y a los suyos el doble de bien. Termino agradeciéndome a mí mismo por haber llegado donde estoy, y donde estaré en el futuro, sea donde sea será bien alto, éxitos para todos ustedes y hasta pronto.

Dedicatoria

A mi madre, la única mujer que nunca me dará la espalda, a ella mi enorme gratitud por ser su hijo y la principal consecuencia de este logro.

El presente trabajo tiene por objetivo la descripción de una propuesta que garantice la administración remota del servicio telemático DHCP en la plataforma Nova-ARST para lograr establecer un control de eficiencia sobre las máquinas de las redes informáticas en la entidad que sea utilizada. Actualmente muchas entidades nacionales realizan este proceso de administración empleando soluciones de pago como Microsoft Windows Server, lo cual limita sustancialmente en cuanto a posibles mejoras a aplicar, soporte, mantenimiento, entre otros factores. La propuesta realizada es consecuente con la estrategia definida por el Consejo de Ministros de comenzar a migrar gradualmente todos los servicios de red nacionales a Software Libre. Teniendo en cuenta esta necesidad, se ha modelado e implementado una plataforma web que permitirá administrar cómoda y profesionalmente el servicio DHCP ya sea de manera local o remota. El sistema permitirá realizar operaciones con las entidades o entradas pertenecientes al servidor, como agregaciones, modificaciones, eliminaciones y búsquedas. Posibilitando además realizar una gestión sobre los ficheros relacionados a este servicio, empleado para el intercambio de información entre servidores-clientes, dicha administración remota será garantizada por configuraciones validadas por la interfaz del usuario sin errores que provoquen un colapso en la red. Podrán visualizarse las opciones disponibles en el módulo del servicio DHCP, habilitados para brindar una secuencia de pasos lógicos y recurrentes que permitan al administrador de red con poca experiencia en la plataforma el uso correcto de este servicio telemático. En términos de portabilidad será posible extender la plataforma a otras versiones del Sistema Operativo GNU/Linux Nova Servidor que sean desarrolladas. Una plataforma que contenga este módulo de este tipo, dadas las condiciones actuales cubanas, es sin lugar a dudas un recurso importante para los administradores de redes. El análisis de factibilidad realizado demuestra que es totalmente viable acometer el proyecto y que la Universidad de las Ciencias Informáticas está dando pasos sólidos en correspondencia con la visión que ha alcanzado el país.

Datos de contacto

Declaro por este medio que yo Reynier Fernández Rodríguez, con carné de identidad 96020422123 soy el autor principal del trabajo titulado “Módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste firman la presente a los __ días del mes de noviembre del año 2022.

Reynier Fernández Rodríguez
Autor

Ing. Profesor. Instructor Hanny Valdés Hernández
Tutora

Ing. Enrique Muschett Cortina
Tutor

Datos del autor

Nombre y apellidos: Reynier Fernández Rodríguez

Correo electrónico: rfrdriguez@estudiantes.uci.cu

Institución a la que pertenece: Universidad de las Ciencias Informáticas.

Dirección: Carretera a San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de la Habana, Cuba, Código Postal 19370.

Datos de los tutores

Nombre y apellidos: Hanny Valdés Hernández

Correo electrónico: hanny@uci.cu

Especialidad de graduación: Ingeniero en Ciencias Informáticas.

Institución a la que pertenece: Universidad de las Ciencias Informáticas.

Dirección: Carretera a San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de la Habana, Cuba, Código Postal 19370.

Nombre y apellidos: Enrique Muschett Cortina

Correo electrónico: emuschett@uci.cu

Especialidad de graduación: Ingeniero en Ciencias Informáticas.

Institución a la que pertenece: Universidad de las Ciencias Informáticas.

Dirección: Carretera a San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de la Habana, Cuba, Código Postal 19370.

Índice	
Introducción	10
Capítulo 1: Fundamentación teórica del módulo de administración remota del servicio telemático DHCP	14
1.1 Conceptos fundamentales	14
Servicio Telemático	14
Administración Remota	15
Protocolo de Internet	16
1.2 Funcionamiento del servicio telemático DHCP	16
1.3 Descripción del proceso de administración del servicio telemático DHCP	18
1.4 Estudio de herramientas que administran de forma remota el servicio DHCP	20
1.4.1 Conclusiones del estudio de homólogos	24
1.5 Metodología de desarrollo del software	24
1.6 Herramientas para el modelado y lenguajes de programación de la propuesta de solución	25
1.6.1 Herramientas de Modelado: UML 2.0	25
1.6.2 Herramienta de modelado: Visual Paradigm 17.0	26
1.6.3 Lenguajes de programación	26
1.6.4 Lenguajes de marcado	27
1.6.5 Hojas de estilo	27
1.6.6 Marcos de trabajo	27
1.6.7 Entornos de desarrollo integrado	28
1.6.8 Librerías	29
1.6.9 Protocolo de seguridad	29
1.7 Conclusiones parciales del capítulo	29
Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST	31
2.1 Propuesta del módulo a desarrollar	31
2.2 Requisitos	32
2.2.1 Fuentes para la obtención de requisitos	32
2.2.2 Técnicas de identificación de requisitos	32
2.2.3 Descripción del módulo DHCP para la plataforma Nova-ARST.. 33	
2.2.4 Especificación de requisitos de software	33
2.2.5 Descripción de requisitos de software	37
2.2.6 Validación de requisitos	38
2.3 Análisis y diseño	39
2.3.1 Diseño arquitectónico	39

2.3.2 Persistencia de datos	41
2.4 Diagrama de paquetes	41
2.5 Patrón de arquitectura React Composition.	41
2.6 Modelo de diseño	42
2.6.1 Diagrama de clases	42
2.7 Conclusiones parciales	47
Capítulo 3. Implementación y pruebas al módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST	48
3.2 Modelo de implementación	49
3.2.1 Diagrama de componentes	49
3.3 Estándar de codificación	51
3.4 Pruebas de software	55
3.5 Aplicación de las pruebas de software	56
3.5.1 Pruebas unitarias.....	56
3.5.2 Pruebas funcionales o de Caja negra	62
3.5.3 Pruebas de aceptación.....	64
3.6 Evaluación del objetivo general de la investigación	65
Resultados obtenidos	68
Cálculo del ISG.....	68
Interpretación del resultado del ISG.....	68
3.7 Interfaz principal	69
3.8 Conclusiones del capítulo	69
Conclusiones	71
Recomendaciones.....	72
Referencias bibliográficas	73
Anexos.....	75

Índice de Tablas

Tabla 1: Resumen de Parámetros por Herramientas. Elaboración propia	23
Tabla 2: Requisitos funcionales del sistema. Elaboración propia	34
Tabla 3: Requisitos no funcionales del sistema. Elaboración propia	36
Tabla 4: Historia de usuario Instalar servicio DHCP. Elaboración propia	37
Tabla 5: Historia de usuario Configurar parámetros del servidor DHCP. Elaboración propia	37
3. Tabla 6: Resultados de las pruebas por casos de prueba. Elaboración propia	60
Tabla 7: Caso de prueba Funcional del RF 1 Instalar servicio DHCP. Elaboración propia	62
Tabla 9: Caso de Prueba de aceptación para la historia de usuario "Instalar servicio DHCP". Elaboración propia	64
Tabla 10: Cuadro lógico de IADOV. Elaboración propia	66

Índice de Figuras

Figura 1: Arquitectura de la plataforma para la administración remota del servicio DHCP. Elaboración propia.	40
Figura 2: Diagrama de paquetes del diseño arquitectónico. Elaboración propia.	41
Figura 3: Diagrama de clases del diseño con estereotipos web de la HU "Editar Interfaz DHCP". Elaboración Propia.....	43
Figura 4: Diagrama de clases del diseño con estereotipos web de la HU "Desinstalar solo el servicio DHCP". Elaboración Propia.....	44
Figura 5: Clase experta en información. Elaboración propia.....	45
Figura 6: Ejemplo código de creación de objeto en la plataforma. Elaboración propia.....	46
Figura 7: Patrón Singleton de la propuesta de solución. Elaboración propia.....	47
Figura 8: Diagrama de despliegue de la plataforma Nova-ARST.....	48
Figura 9: Diagrama de componentes del módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST. Elaboración Propia.	50
Figura 10: Aplicación de los estándares de codificación. Elaboración propia.....	55
Figura 11: Código de implementación para el método del camino básico. Elaboración propia.....	57
Figura 12: Grafo resultante del método caja blanca camino básico. Elaboración propia.....	58
Figura 13: No Conformidades detectadas al aplicar el método de caja negra. Elaboración propia.....	64
Figura 14: Resultados de las pruebas de aceptación. Elaboración propia.....	65
Figura 15: Interfaz principal del módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST. Elaboración propia.....	69

Introducción

Los servicios telemáticos son aquellos que, utilizando como soporte servicios básicos, permiten el intercambio de información entre terminales con protocolos establecidos para sistemas de interconexión abiertos. (MinTIC, 2021). Los servicios telemáticos están presentes en multitud de actividades rutinarias de hoy en día, algunos de los más habituales son servicios de televisión y radio, datos de telefonía celular, asistentes de voz, correos electrónicos, etc. Se trata de soluciones tecnológicas que usan computadores conectados en red para ofrecer un gran conjunto de datos e información. (DispatchTrack, 2020).

Para que esta red funcione de forma organizada existen protocolos. Los protocolos son el conjunto de reglas que las computadoras deben seguir y respetar para poder comunicarse (EDteam, 2022). El protocolo DHCP viene de inglés '*Dynamic Host Configuration Protocol*' el cual pertenece a la capa de aplicación del modelo de referencia de Interconexión de Sistemas Abiertos (OSI, *Open System Interconnection*), se trata de un protocolo de red tipo cliente / servidor que se encarga de asignar direcciones IP de forma dinámica, así como otros parámetros relativos a la configuración de red a cada uno de los dispositivos conectados (AdlsZone, 2022).

El protocolo DHCP le permite al administrador supervisar y distribuir de forma centralizada las direcciones IP necesarias y, automáticamente, asignar y enviar una nueva IP si fuera el caso en que el dispositivo es conectado en un lugar diferente de la red (Optional Computer, 2022). La gestión inteligente se lleva a cabo gracias al servidor DHCP, que cuenta con una lista de direcciones IP dinámicas para poder ir asignándoselas a los clientes a medida que van quedando libres, y considerando siempre cuánto tiempo la tienen, qué cliente la tiene y a cuál se le ha asignado a posteriori (AdlsZone, 2022).

Para eliminar la dependencia de los Sistemas Operativos (SO) de pago Cuba se apoya en la Universidad de Ciencias de la Informática (UCI), surgida con el propósito de informatizar la sociedad cubana, con énfasis en la producción de software. Dentro de su infraestructura docente-productiva se encuentra el Centro de Software Libre (CESOL), adscrito a la Facultad 1, el cual desarrolla la Distribución Cubana de GNU/Linux, Nova Servidor (SO libre, basado en Ubuntu). Igualmente conduce los procesos de migración a aplicaciones de código abierto, desde un modelo de integración de la formación, la investigación y el postgrado, contribuyendo a la formación integral de profesionales comprometidos con la Revolución que respondan a las necesidades del progreso científico - técnico y socio - económico del país (Portal Web UCI, 2022).

Dentro de los procesos de migración a aplicaciones de código abierto en la Distribución Cubana de GNU/Linux, Nova Servidor se encuentran el servicio telemático DHCP, el mismo se realiza manualmente a través de una interfaz de línea de comandos, requiriendo una gran cantidad de operaciones. Esto provoca dificultades tales como lentitud durante la ejecución de las tareas, aumento de la complejidad en el proceso de migración, mayor cobertura a la introducción de errores humanos e incluso la pérdida de información. Hoy en día GNU/Linux Nova no cuenta con una interfaz que abarque todos los servicios DHCP de forma conjunta y centralizada, para la administración de servidores DHCP en una red de cualquier entidad en nuestro país.

Debido a esto todos los procesos de instalación, configuración y administración de los servicios DHCP se encuentran dispersos y solo se pueden acceder a su ejecución a través de consola, provocando que no se ejecuten en el momento deseado y por consiguiente, atentan contra el éxito de la migración de los datos y servicios. Muchas de las empresas donde se instala este SO cuentan con varios servidores DHCP lo que hace engorroso el trabajo a todos los administradores de red que se encargan de la seguridad y cuidado de la información que se manda de un computador a otro, teniendo que administrar estos servidores por separado.

En correspondencia con lo antes expuesto se plantea como **problema de investigación:** ¿Cómo facilitar la administración remota del servicio DHCP desde la plataforma Nova-ARST?

Se identifica como **objeto de investigación:** el proceso de administración del servicio telemático DHCP. Quedando enmarcado el **campo de acción** en la administración remota del servicio telemático DHCP.

Para darle solución al problema anteriormente planteado, se define como **objetivo general:** desarrollar un módulo para la administración remota del servicio telemático DHCP desde la plataforma Nova-ARST en la Distribución Cubana GNU/Linux Nova.

Para dar cumplimiento al objetivo general se han definido los siguientes **objetivos específicos:**

1. Elaborar el marco teórico de la investigación sobre el proceso de administración del servicio telemático DHCP.
2. Diseñar un módulo para Nova-ARST que permita la administración remota del servicio telemático DHCP.
3. Implementar un módulo para Nova-ARST que permita la administración remota del servicio telemático DHCP.

4. Evaluar el módulo para Nova-ARST que permita la administración remota del servicio telemático DHCP.

Se definen como **preguntas científicas**:

1. ¿Cuáles son los presupuestos teóricos que fundamentan el proceso de administración remota del servicio telemático DHCP en Nova-ARST?
2. ¿Qué aspectos se deben tener en cuenta para diseñar un módulo que permita la administración remota del servicio telemático DHCP en Nova-ARST?
3. ¿Qué componentes se deben implementar para obtener un módulo que permita la administración remota del servicio telemático DHCP en Nova-ARST?
4. ¿Qué métodos, técnicas y pruebas aplicar para la evaluación del módulo que permita la administración remota del servicio telemático DHCP en Nova-ARST?

Para facilitar el cumplimiento del objetivo propuesto y de las tareas de investigación se emplean los siguientes métodos científicos:

Métodos teóricos

Histórico-Lógico: Se utiliza el método histórico para buscar toda la información relacionada con el servicio telemático DHCP en sistemas operativos en la actualidad. Se emplea para conocer las características de otras plataformas que puedan tener funciones similares a la que se desea implementar a través de un estudio de homólogos. Se maneja el método lógico para interpretar toda la información y concluir con una plataforma que encierre las funcionalidades necesarias que se adapten a la necesidad de administrar de forma remota el servicio telemático DHCP en cualquier empresa del país.

Analítico-Sintético: Se hace uso del método analítico para describir cómo podría ser una administración remota del servicio telemático DHCP desde una plataforma que abarque todos los elementos necesarios que garanticen su funcionamiento acorde a lo que se necesita y así descomponer los elementos por separado del problema de investigación para conocer, explicar y relacionar las características de una administración remota y un servicio telemático DHCP. Se hace provecho el método sintético para formular una teoría que unifique los elementos dispersos del servicio DHCP a través de diagramas y modelos que garanticen una administración remota.

Inductivo-Deductivo: Se emplea el método inductivo para arribar a conocimientos sobre el proceso de instalación, configuración y gestión del servicio telemático DHCP de forma remota y así encontrar los rasgos comunes para llegar a conclusiones de los aspectos que los caracterizan. Se aplicó el método deductivo para llegar a la conclusión

del uso de una plataforma que incluya las características del servicio telemático DHCP que faciliten el trabajo de un administrador de redes en el SO GNU/Linux Nova.

Métodos empíricos

Entrevista: El investigador hizo contacto personal con algunos especialistas de CESOL con experiencia en la administración de redes que propició reconocer aspectos afectivos y volitivos que pueden ser relevantes para la investigación. Se hizo de forma individual a través de una serie de preguntas abiertas al consultado con el fin de que su respuesta pueda ofrecer la información que se persigue buscar (Anexo 1).

El presente trabajo de diploma consta con una introducción, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas y anexos, cuya estructura se describe a continuación:

Capítulo 1: Fundamentación teórica del módulo de administración remota del servicio telemático DHCP

En este capítulo se explican los principales conceptos que permiten lograr un mayor entendimiento del tema tratado. Se realiza un estudio sobre la administración remota de los servicios telemáticos y las herramientas que los implementan. Además, se definen las tecnologías que se van a utilizar para el desarrollo del módulo de administración remota del servicio telemático DHCP para la plataforma Nova-ARST.

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

En este capítulo se especifican los requisitos tanto funcionales como no funcionales para lograr que el módulo de administración remota del servicio telemático DHCP se integre correctamente a la plataforma Nova-ARST. También se describen las historias de usuario correspondientes a los requisitos funcionales, la arquitectura y los patrones de diseño que se utilizan durante la implementación.

Capítulo 3: Implementación y pruebas al módulo de administración remota del servicio DHCP en la plataforma Nova-ARST

En este capítulo se muestra el modelo de implementación como resultado del diseño anteriormente desarrollado, así como el diagrama de componentes, y se describen las pruebas a realizar, con el objetivo de comprobar el correcto funcionamiento del módulo de administración remota del servicio telemático DHCP para la plataforma Nova-ARS

Capítulo 1: Fundamentación teórica del módulo de administración remota del servicio telemático DHCP

Capítulo 1: Fundamentación teórica del módulo de administración remota del servicio telemático DHCP

En el presente capítulo se definen los conceptos fundamentales que respaldan la investigación. Se fundamenta el objeto de estudio y el campo de acción de la plataforma a desarrollar. Se hace un análisis en el mercado de las herramientas homólogas para obtener una comparación entre las mismas y arribar al desarrollo de una plataforma adaptable y beneficiosa al país. Se realiza un análisis sobre la administración del servicio telemático DHCP y las ventajas de su administración remota. Además, se definen las herramientas, tecnologías y la metodología de desarrollo de software a utilizar en la propuesta de solución.

1.1 Conceptos fundamentales

Para lograr un mejor entendimiento del problema a investigar se tuvieron en cuenta conceptos importantes asociados al objeto de estudio de la investigación: servicio telemático, administración remota, Protocolo de Internet.

Servicio Telemático

Se divide para una mayor comprensión servicio telemático en servicio y telemática obteniéndose según el Diccionario RAE (Real Academia Española) como “prestación que satisface alguna necesidad humana y que no consiste en la producción de bienes materiales”. Un servicio es una prestación, un activo de naturaleza económica pero que no tiene presencia física propia (es intangible), a diferencia de los bienes que si la tienen (Software Delsol, 2022). La palabra telemática se refiere a la combinación de la informática y de la tecnología de la comunicación para el envío y recepción de datos. En sentido amplio, resulta de la unión de las telecomunicaciones y la informática, pudiendo implicar desde la tecnología de envío, recepción y almacenamiento de información a través de dispositivos de telecomunicación (Viasat, 2020).

Los servicios telemáticos se basan en la disponibilidad de una infraestructura de red capaz de soportarlos, sin ella la mayoría de estos no se pueden ofrecer. En todos los países se están creando estas infraestructuras de comunicaciones capaces de presentar los servicios que los ciudadanos demandan, pero cuando descienden a niveles de empresas, suelen ser las mismas las que se encargan de crear lo que se denomina red corporativa, destinada a dar servicio a la entidad que se trate. Hoy las comunicaciones de empresa incorporan voz, datos, textos e imagen, y necesitan de altas prestaciones, flexibilidad, disponibilidad y calidad, además de una alta relación coste servicio.

Capítulo 1: Fundamentación teórica del módulo de administración remota del servicio telemático DHCP

Existe gran diversidad de **ejemplos de servicios telemáticos**. Algunos de los más habituales son los siguientes:

- ✓ Asistentes de voz: los asistentes de voz como Alexa o Siri se conectan a Internet para obtener la información necesaria y, de este modo, ofrecer los servicios solicitados por los usuarios.
- ✓ Servicios de TV y radio: dos **ejemplos de telemática** convencionales son tanto la TV como la radio. Hoy en día, estos medios emiten su señal de forma tradicional, pero también utilizando otros formatos. Por ejemplo, mediante Internet.
- ✓ Redes profesionales y académicas: las redes que se utilizan en entornos laborales y educativos usan tecnología **telemática** para mantener conectados a todas las partes que trabajan y estudian de forma conjunta.
- ✓ Rastreo satelital con GPS: los sistemas de posicionamiento satelital son también uno de los **ejemplos de telemática** actuales más usados en el día a día de multitud de personas, especialmente en el sector de la logística.
- ✓ Datos de telefonía celular: los celulares actuales son uno de los mejores **ejemplos de servicios telemáticos**. Se trata de teléfonos que, al conectarse a Internet, comparten constantemente información y mantienen actualizado al usuario en todo momento.
- ✓ Rastros inteligentes: usando la tecnología IoT, se puede conocer la posición exacta y en tiempo real de determinados objetos o elementos conectados (DispatchTrack, 2022).

Administración Remota

La administración remota es el proceso que implica la gestión, dirección y control de los recursos de una empresa desde un local a una cierta distancia retirada o alejada. Es la acción de controlar una computadora u otro dispositivo a través de otro, que puede encontrarse en un punto geográfico diferente. Esto quiere decir que una persona puede utilizar un ordenador en La Habana, por ejemplo, y acceder a otro que está en Granma a través de un programa y una conexión a Internet. Al conectarse con el aparato remoto, observará en pantalla su espacio de trabajo (que puede ser su escritorio) y podrá interactuar con sus programas y documentos, tal y como si estuviera allí. Una de las aplicaciones principales del acceso remoto es la asistencia técnica, ya que permite a los especialistas llevar a cabo el mantenimiento de un gran número de dispositivos sin

Capítulo 1: Fundamentación teórica del módulo de administración remota del servicio telemático DHCP

moverse de su lugar de trabajo, y beneficia a los usuarios ofreciéndoles la posibilidad de recibir ayuda inmediata ante una emergencia. (Definición.de, 2008).

Protocolo de Internet

El Protocolo de Internet (IP, de sus siglas en inglés Internet Protocol) es un protocolo NO orientado a conexión usado tanto por el origen como por el destino para la comunicación de datos a través de una red de paquetes conmutados.

Los datos en una red que se basa en IP son enviados en bloques conocidos como paquetes o datagramas (en el protocolo IP estos términos se suelen usar indistintamente). En particular, en IP no se necesita ninguna configuración antes de que un equipo intente enviar paquetes a otro con el que no se había comunicado antes.

El Protocolo de Internet provee un servicio de datagramas no fiable (también llamado del mejor esfuerzo (best effort), lo hará lo mejor posible pero garantizando poco). IP no provee ningún mecanismo para determinar si un paquete alcanza o no su destino y únicamente proporciona seguridad (mediante checksums o sumas de comprobación) de sus cabeceras y no de los datos transmitidos. Por ejemplo, al no garantizar nada sobre la recepción del paquete, éste podría llegar dañado, en otro orden con respecto a otros paquetes, duplicado o simplemente no llegar. Si se necesita fiabilidad, ésta es proporcionada por los protocolos de la capa de transporte, como TCP (Protocolo de Control de Trasmisión, por sus siglas en inglés Transmission Control Protocol)

Si la información a transmitir ("datagramas") supera el tamaño máximo "negociado" (MTU) en el tramo de red por el que va a circular podrá ser dividida en paquetes más pequeños, y re ensamblada luego cuando sea necesario. Estos fragmentos podrán ir cada uno por un camino diferente dependiendo de cómo estén de congestionadas las rutas en cada momento.

Las cabeceras IP contienen las direcciones de las máquinas de origen y destino (direcciones IP), direcciones que serán usadas por los enrutadores (routers) para decidir el tramo de red por el que reenviarán los paquetes (LWP, 2022).

1.2 Funcionamiento del servicio telemático DHCP

El protocolo DHCP es habitual en las redes IP hoy en día ya que nos permite auto automatizar el proceso de configuración TCP/IP de un equipo de tal modo que facilita conexión de nuevos equipos a la red. El proceso de conexión de un equipo a una red , sin importar que del tipo que sea impresora, servidor, teléfono o videoconsola cuando lo conectamos a una red , bien sea por cable o wifi para poder comunicarnos con otros elementos de la red se necesita algo más que el cable, algo que no sea físico , es decir lógico que se configura en el equipo, estamos hablando en la configuración TCP/IP, a lo

Capítulo 1: Fundamentación teórica del módulo de administración remota del servicio telemático DHCP

largo de los años se ha aplicado de forma manual tanto en dispositivos de red como routers y switches o dispositivos finales como PC y servidores. Pero también existe una manera automática con el uso del protocolo DHCP sin tener que hacer nada, permite que el equipo se encargue de iniciar un proceso en el que acabara obteniendo la configuración TCP/IP y así podrá comunicarse con otros elementos sin la necesidad de que nadie tenga que ir al apartado configuración del sistema operativo e introducir los diferentes valores de IP, mascara de subred y puerta de enlace, cuando hablamos de una red doméstica no supone mucha diferencia en cambio si el escenario es una red mediana o grande como el de una empresa o corporación se hace más engorroso.

Tiene ventajas como el aprovechamiento del espacio de direcciones IP, facilitar la movilidad/conectividad y minimiza errores de configuración. Al ser un protocolo de la capa de aplicación, existen dos roles el cliente y el servidor, el cliente que espera recibir la información de configuración TCP /IP y por otro lado el servidor que se encargue de proporcionar esta información al cliente, ambos deben estar conectados de algún modo a red. En cuanto al servidor debe tener instalado este servicio del servidor DHCP es un servicio muy habitual en servidores Windows o Linux que se puede instalar y habilitar siempre que se desee.

Cuando conectamos un equipo a la red y este tiene habilitado el DHCP en su configuración TCP/IP de red iniciara un proceso de 4 fases que por sus siglas se llaman DORA (Discover, Offer, Request, Acknowledge) en español (Descubrir, Oferta, Solicitud, Reconocer).

Primero manda un mensaje al servidor del tipo Discover en el que va intentar descubrir si existe un servidor DHCP en la red para que este le ofrezca la información que necesita , es una forma de decirle a la red , necesito una dirección IP y este mensaje Discover es importante que llegue a todos los elementos de la red, este mensaje se envía a través de un broadcast que contiene la MAC destino de la trama (dirección broadcast: ffff.ffff.ffff.ffff) cuando llegue al servidor DHCP este lo hará en base a la información que tiene configurada para el servicio DHCP , donde buscara una dirección IP que este libre y respondería (Offer) con un mensaje DHCP en el que indicara toda la configuración TCP/IP que le ofrece al cliente , agregado un parámetro llamado Lease Time (en español Tiempo de Arrendamiento) que le indicara al cliente el tiempo que podrá usar esta dirección IP (el aprovechamiento del espacio de direcciones IP) y así no ocupar eternamente con concesiones antiguas. Como tercer paso el cliente responderá con un mensaje (Request) donde confirma al servidor DHCP que quiere asignarse esta dirección IP concedida esta dirección IP será a través de la trama llamada

Capítulo 1: Fundamentación teórica del módulo de administración remota del servicio telemático DHCP

UNICAST. Para acabar el proceso el servidor responderá con un Acknowledge confirmando la asignación de esta dirección IP.

1.3 Descripción del proceso de administración del servicio telemático DHCP

Todas las direcciones IP de todos los equipos se almacenan en una base de datos que reside en un servidor. Un servidor DHCP puede proporcionar los ajustes de configuración utilizando dos métodos.

Rango de Direcciones

Este método se basa en la definición de un grupo de las direcciones IP para los clientes DHCP (*también llamado IP address pool*) que suministran sus propiedades de configuración de forma dinámica según lo soliciten los ordenadores cliente. Cuando un cliente DHCP ya no está en la red durante un período determinado, la configuración vence y la dirección IP del *poll* (encuesta) es puesta en libertad el uso de otros clientes DHCP.

Dirección MAC

Este método se basa en utilizar el protocolo DHCP para identificar la dirección de hardware única de cada tarjeta de red conectada a la red y luego es asignada una configuración constante, así como la misma dirección IP cada vez que la configuración de DHCP del cliente realiza una petición al servidor DHCP desde el mismo dispositivo de red.

Instalar un servicio DHCP en GNU/Linux Nova

Para instalar el servidor de asignación automática de dirección IP ejecutamos el comando:

```
sudo apt-get install isc-dhcp-server
```

Configurar el servidor DHCP en GNU/Linux Nova

En el caso que tengan dos interfaces de red *NIC* en su servidor Linux tienen que seleccionar cual van a utilizar para escuchar las peticiones DHCP. Para configurar el servicio, editamos el archivo `/etc/default/dhcp3-server`, y cambiamos `INTERFACES="eth0"` por la tarjeta de red interna.

Es necesario hacer una copia de seguridad del archivo de configuración:

```
cp /etc/dhcp3/dhcpd.conf /etc/dhcp3/dhcpd.conf.back
```

Configurar utilizando el método de rango de direcciones (IP pool)

Editamos la configuración tecleando:

Capítulo 1: Fundamentación teórica del módulo de administración remota del servicio telemático DHCP

```
sudo vi /etc/dhcp3/dhcpd.conf
```

Y en este archivo cambiamos las siguientes secciones

```
default-lease-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.1;
option domain-name-servers 192.168.1.9, 192.168.1.10;
option domain-name "guatewireless.org";
subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.10 192.168.1.200;
}
```

Guardamos y salimos del archivo. El texto anterior configura el servidor DHCP con los siguientes parámetros:

- ✓ Asignación a los clientes direcciones IPs del rango de 192.168.1.10 hasta 192.168.1.200
- ✓ Prestará la dirección IP por un mínimo de 600 segundos, y como máximo permitido de 7200 segundos.
- ✓ Determina la máscara de subred a 255.255.255.0
- ✓ Dirección de broadcast de 192.168.1.255
- ✓ Como gateway/pasarela de red/router la dirección 192.168.1.1
- ✓ Y los servidores 192.168.1.9 y 10 como sus servidores DNS

Reservas de IP para ciertas máquinas

Para esto al final de archivo agregamos el siguiente bloque

```
# SERVER
host SERVER {
hardware ethernet xx:xx:xx:xx:xx:xx;
fixed-address 192.168.1.210;
option broadcast-address 192.168.0.255;
}
```

En donde *SERVER* es el nombre de la máquina, *xx:xx:xx:xx:xx:xx* es la *mac address* de la máquina, 192.168.1.210 es la IP que queremos reservar y 192.168.0.255 es la dirección de *broadcast*.

Capítulo 1: Fundamentación teórica del módulo de administración remota del servicio telemático DHCP

Guardamos el archivo reiniciamos el servidor dhcp usando.

```
/etc/init.d/dhcp3-server restart
```

Configurar utilizando el método de direcciones MAC

Con este método se puede **reservar** algunas o todas las direcciones IP de nuestra red para determinadas maquinas. Como podrán ver la configuración es muy parecida a la anterior, con la salvedad que para reservar la asignación de una IP a una determinada NIC (*network card interface*) debemos de utilizar la etiqueta *host*

```
default-lease-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.1;
option domain-name-servers 192.168.1.9, 192.168.1.10;
option domain-name "guatewireless.org";
subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.10 192.168.1.200;
}
host oracle{
hardware ethernet 00:03:47:31:e1:7f;
fixed-address 192.168.1.20;
}
host printer {
hardware ethernet 00:03:47:31:e1:b0;
fixed-address 192.168.1.21;
}
```

Ahora reiniciamos el servidor dhcp ejecutando el siguiente comando:

```
sudo /etc/init.d/dhcp3-server restart
```

1.4 Estudio de herramientas que administran de forma remota el servicio DHCP

Webmin

Capítulo 1: Fundamentación teórica del módulo de administración remota del servicio telemático DHCP

Webmin es una herramienta de configuración de sistemas, accesible vía web para OpenSolaris, GNU/Linux y otros sistemas Unix, la cual permite configurar aspectos internos de muchos sistemas operativos, como gestión de usuarios, cuotas de espacio, servicios, archivos de configuración y apagado del equipo. Está escrita en Perl y por defecto se comunica mediante el protocolo TCP a través del puerto 10000. Con esta herramienta se pueden configurar los permisos para Usuarios y Grupos o configurar el funcionamiento del servidor Mysql Server, cuotas de espacio, servicios, archivos de configuración, apagado del equipo, entre otras opciones. Webmin tiene una estructura por módulos para administrar una amplia variedad de herramientas como Apache, PHP, MySQL, DNS, Samba, DHCP, Proxy, entre otros, además de ser completamente configurables y de poder crear nuevos módulos.

Funcionalidades que brinda

Este servidor administra el servicio proxy a través de Squid, permitiendo las siguientes configuraciones:

- ✓ Establecer las configuraciones básicas y avanzadas para el correcto funcionamiento del servicio como puerto de escucha, parámetros de la cache, reglas de acceso.
- ✓ Sincronización del servicio con el cortafuego del sistema.
- ✓ Administrar opciones de registros del sistema.
- ✓ Reconoce los cambios realizados en el fichero principal de configuración.

Zentyal

Zentyal (anteriormente conocido como *eBox Platform*) es un servidor de red unificada de código abierto (o una plataforma de red unificada) para las pequeñas y medianas empresas. Zentyal puede actuar gestionando la infraestructura de red, como puerta de enlace a Internet (Gateway), gestionando las amenazas de seguridad (UTM), como servidor de oficina, como servidor de comunicaciones unificadas o una combinación de estas. Además, Zentyal incluye un marco de desarrollo (un framework) para facilitar el desarrollo de nuevos servicios basados en Unix. Para la administración del servicio proxy este servidor utiliza Squid permitiendo las siguientes configuraciones:

- ✓ Definir si el proxy funcionará en modo transparente.
- ✓ Filtrado de páginas Web según su contenido.
- ✓ Implementar un límite flexible para controlar el ancho de banda que consumen los usuarios.
- ✓ Configuración de los parámetros básicos de administración del servicio proxy como puerto escucha, cache, almacén central de librerías y validación de usuario.

Capítulo 1: Fundamentación teórica del módulo de administración remota del servicio telemático DHCP

El módulo de Zentyal para la administración del servicio proxy presenta una interfaz con un alto grado de usabilidad. No reconoce y sobre escribe las configuraciones establecidas en el fichero principal de configuración de Squid, de manera que no permitirá realizar configuraciones que no estén establecidas en el módulo como las comunicaciones entre proxy. Realiza la administración del servicio FTP de forma rígida, ya que en su interfaz contempla solamente configuraciones básicas, realizando otras predeterminadamente, sin posibilidad de ser modificadas, a no ser, mediante el acceso directo al fichero ubicado en /etc/vsftpd.conf. Algunos de estos parámetros son la activación de registros, la máscara de los archivos cuando son creados, tanto para usuarios locales como anónimos, el texto de bienvenida y la activación de los mensajes de directorio. No tiene en cuenta directivas como la cantidad máxima de usuarios e IP conectados de forma simultánea, el tiempo máximo de conexiones inactivas, configuración de usuarios virtuales y máxima velocidad de transferencia de datos para usuarios locales y anónimos. No ofrece un manual de ayuda detallado para realizar las configuraciones, solamente cuenta con una breve explicación sobre el acceso y los permisos de los usuarios anónimos.

HMAST

HMAST es el sistema que permite administrar los servidores de forma remota, el cual tiene las funcionalidades necesarias para la gestión de usuarios, tareas programadas y servicios, desarrollada por en la Facultad 1 de la Universidad a partir del año 2013. Permite la administración de conexiones a través del protocolo SSH y así como la del servicio DHCP. HMAST cuenta con un sistema base que permite la administración de las PC servidoras de forma remota, el cual tiene las funcionalidades necesarias para administrar los usuarios, las tareas programadas y los servicios, entre otros.

Arquitectura del sistema

La arquitectura que presenta HMAST propone el diseño de una arquitectura N-Capas basada en el dominio compuesta por cinco capas.

Funcionalidades que brinda

- ✓ Gestión de servidores lógicos: Permite la adición, edición, eliminación de los datos de un servidor lógico, además permite la conexión remota y desconexión a un servidor seleccionado.
- ✓ Gestión de servicios telemáticos asociados a un servidor lógico: Permite la adición, edición, eliminación de los datos de un módulo, así como activación y desactivación de los mismos.

Capítulo 1: Fundamentación teórica del módulo de administración remota del servicio telemático DHCP

- ✓ Gestión de los variables de configuración asociadas a un servidor lógico: Permite cargar y salvar las variables de configuración de los servicios telemáticos encontrado en un servidor lógico (ficheros de configuración, nombre de módulos, demonios, entre otros).

Para una mayor comprensión de las plataformas o herramientas antes descritas se tomó en cuenta algunos parámetros de comparación entre las mismas que arriben a una conclusión más exacta de elementos que debe o no debe tener la plataforma a desarrollar. Para determinar estos parámetros se realizó en apoyo del documento Guía para la evaluación y comparación de software ¿Cómo evaluar y comparar software? (Agesic, versión 1.7, 2019).

Tabla 1: Resumen de Parámetros por Herramientas. Elaboración propia

Parámetros	Webmin	Zentyal	HMAST
Costo de instalación	Depende de paquetes bajo licencia	Depende de paquetes bajo licencia	Depende de paquetes disponibles en repositorio nacional.
Dependencia	Linux y Solvetic	GNU/Linux Ubuntu	GNU/Linux Nova 5
Tipo de Licencia	Licencia BSD	Licencia LGPL	Disponible de forma gratis para el país
Migración de datos	Complejidad Alta	Complejidad Media	Complejidad Media
Performance o interfaz de usuario	Aceptable	Aceptable	Aceptable
Funcionalidades remotas del módulo DHCP	Si	Si	No
Actualizaciones y mantenimiento	Dispone de actualización hasta la fecha 18-sept-2022	Dispone de actualización hasta el 26 de enero de 2021	Dispone de actualización hasta el año 2016
Facilidad de Uso	Alta	Media	Media
Seguridad	Alta	Alta	Alta

Capítulo 1: Fundamentación teórica del módulo de administración remota del servicio telemático DHCP

Flexibilidad	Alta	Media	Baja
--------------	------	-------	------

1.4.1 Conclusiones del estudio de homólogos

La tabla que se presenta anteriormente nos informa de aspectos positivos y negativos a la hora de desarrollar más adelante una plataforma que absorba los mejores aspectos de estas herramientas. Se debe tener en cuenta según la tabla anterior que a pesar de Webmin y Zentyal tener relación con SO de código abierto como GNU/Linux no tiene relación con el SO Nova Servidor 8.0 basado en GNU/Linux Ubuntu, siendo no atractivas a las necesidades de las empresas cubanas que hacen uso de este SO.

Webmin y Zentyal tienen licencias BSD y LGPL respectivamente lo que en algunos casos no tienen autorizado el uso de su código fuente o en caso de ser modificados deben ser publicados en la comunidad, no siendo un aspecto positivo para el desarrollo de la futura plataforma. En el caso de HMAST es una herramienta desarrollada en nuestra Universidad, pero dejó de ser actualizada después de la emigración de datos de Nova 5.0 a versiones más avanzadas y su actualización requiere de paquetes que a pesar de estar disponibles modificarlos sería de mayor costo en tiempo, personal capacitado y paquetes relacionados con GNU/Linux en versiones más antiguas. Se debe tomar en cuenta a la hora de desarrollar la nueva plataforma la flexibilidad o capacidad de adaptación de futuras funcionalidades, seguridad y facilidad de uso de Webmin, no así la interfaz de usuario que puede llegar a ser compleja a la vista de un administrador de redes y seguridad informática. La plataforma debe contar con un módulo DHCP con administración remota como lo tiene Webmin y HMAST.

1.5 Metodología de desarrollo del software

Una metodología es un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo. Es un proceso de software detallado y completo. La metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito, comprende los procesos a seguir sistemáticamente para idear, implementar y mantener un producto de software desde que surge la necesidad del producto hasta que se cumple el objetivo por el cual fue creado (Maida, 2015). A continuación, se describe la metodología de desarrollo de software que se va a emplear para llevar a cabo la presente investigación.

Variación de AUP para la UCI es una variación de la metodología Proceso Unificado Ágil (AUP, del inglés *Agile Unified Process*), esta se adapta al ciclo de vida definido para la actividad productiva de la UCI. Esta metodología se apoya en la Integración de

Capítulo 1: Fundamentación teórica del módulo de administración remota del servicio telemático DHCP

Modelos de Madurez de Capacidades para Desarrollo (CMMI-DEV, del inglés *Capability Maturity Model Integration Development*). El cual constituye una guía para aplicar las mejores prácticas en una entidad desarrolladora (Rodríguez, 2014). La metodología cuenta con las siguientes fases:

Inicio: se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.

Ejecución: en esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.

Cierre: en esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Esta metodología se adapta al ambiente de trabajo y le permite al cliente acompañar al equipo de desarrollo para convenir los requisitos y así poder implementarlos. El desarrollo de la investigación se centrará en la fase de Ejecución y transitará por las siguientes disciplinas: requisitos, análisis, diseño, implementación, pruebas internas y pruebas de aceptación, definidas en la metodología.

El escenario escogido para esta investigación es el número cuatro, pues la solución no posee un negocio, este escenario se utiliza para modelar el sistema a través de HU (Historia de Usuario). Además, se puede concluir que esta metodología se ajusta a cualquier proyecto productivo de la UCI.

1.6 Herramientas para el modelado y lenguajes de programación de la propuesta de solución

Para llevar a cabo el desarrollo del módulo de administración remota del servicio telemático en la plataforma, dar cumplimiento a los objetivos de esta investigación, y teniendo en cuenta las características del producto que se desea obtener; es necesario realizar una descripción de los lenguajes y tecnologías implicadas en el proceso de desarrollo.

1.6.1 Herramientas de Modelado: UML 2.0

El Lenguaje Unificado de Modelado (UML, del inglés *Unified Modeling Language*) es un lenguaje de modelado visual que permite especificar, construir y documentar artefactos de un software. Se puede usar en las diferentes etapas del ciclo de vida de un proyecto. Incluye conceptos semánticos, notación y principios generales de un sistema. Contiene

Capítulo 1: Fundamentación teórica del módulo de administración remota del servicio telemático DHCP

además construcciones organizativas para agrupar los modelos en paquetes, lo que permite dividir grandes sistemas en piezas de trabajo más simples. Este lenguaje es lo suficientemente expresivo como para modelar sistemas que no son informáticos, como flujos de trabajo en una empresa, diseño de la estructura de una organización y el diseño del hardware. Un modelo UML está compuesto por tres clases de bloques de construcción (Lucidchart, 2017):

- Elementos: los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etcétera).
- Relaciones: relacionan los elementos entre sí.
- Diagramas: son colecciones de elementos con sus relaciones.

1.6.2 Herramienta de modelado: Visual Paradigm 17.0

Visual Paradigm es una herramienta de Ingeniería de Software Asistida por Computación (CASE, del inglés Computer Aided Software Engineering). Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas (Pressman, 2002).

Características:

- Disponibilidad en múltiples plataformas (Windows, Linux).
- Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento.

1.6.3 Lenguajes de programación

Un lenguaje de programación consiste en un vocabulario que contiene un conjunto de reglas gramaticales destinadas a girar instrucciones a un ordenador o dispositivo informático para que ejecuten tareas específicas (Chakray, 2015).

Python 3.8

Python es un lenguaje de programación potente y fácil de aprender. Tiene estructuras de datos de alto nivel eficientes y un enfoque simple pero efectivo para la programación orientada a objetos. La sintaxis elegante y la tipificación dinámica de Python, junto con su naturaleza interpretada, lo convierten en un lenguaje ideal para secuencias de comandos y desarrollo rápido de aplicaciones en muchas áreas en la mayoría de las plataformas.

Capítulo 1: Fundamentación teórica del módulo de administración remota del servicio telemático DHCP

Se selecciona el lenguaje de programación Python, debido a que es el utilizado por la plataforma de administración remota de servicios telemáticos Nova-ARST a la que se debe integrar la presente propuesta de solución, por lo que su utilización contribuye a la homogeneidad y compatibilidad del código.

Java Script 5.1

JavaScript es un lenguaje de programación que usa script (rutinas o guiones) usado para crear interactividad dinámica en los sitios web. Es un lenguaje multi-paradigma basado en prototipos dinámicos. Las capacidades dinámicas de JavaScript incluyen construcción de objetos en tiempo de ejecución. JavaScript es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Opera, Mozilla Firefox, entre otros. Centrado en el código fuente de la página web (Mdn, 2017).

1.6.4 Lenguajes de marcado

HTML5 (Hyper Text Markup Language, Lenguaje de Marcado de Hipertexto) es un lenguaje de programación utilizado para estructurar y presentar el contenido en la web. Define los nuevos estándares de desarrollo web, rediseñando el código para resolver determinados problemas. No se limita solo a crear nuevas etiquetas o atributos, sino que incorpora muchas características nuevas y proporciona una plataforma de desarrollo para complejas aplicaciones web (Garro, 2015).

1.6.5 Hojas de estilo

CSS u hojas de estilo en cascada (en inglés *Cascading Style Sheets*) es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML. Es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Este lenguaje es usado para definir los estilos de los contenidos, es decir, el color, tamaño y tipo de letra de los párrafos de texto, la tabulación con la que se muestran los elementos de una lista y la separación entre titulares y párrafos para agregar belleza, diseño, orden a las páginas web ya estructuradas con HTML y así hacer más agradable la experiencia de uso de una página web (Navaja, 2012).

1.6.6 Marcos de trabajo

Django 2.2

Django es un marco web Python de alto nivel que fomenta un desarrollo rápido y un diseño limpio y pragmático. Creado por desarrolladores experimentados, se ocupa de gran parte de las molestias del desarrollo web, por lo que puede concentrarse en escribir su aplicación sin necesidad de reinventar la rueda. Es gratis y de código abierto.

Capítulo 1: Fundamentación teórica del módulo de administración remota del servicio telemático DHCP

Incluye docenas de extras que puede usar para manejar tareas comunes de desarrollo web. Se encarga de la autenticación del usuario, la administración de contenido, los mapas del sitio, las fuentes RSS y muchas más tareas, desde el primer momento. Se toma la seguridad muy en serio y ayuda a los desarrolladores a evitar muchos errores de seguridad comunes, como la inyección de SQL, las secuencias de comandos entre sitios, la falsificación de solicitudes entre sitios y el secuestro de clics. Su sistema de autenticación de usuarios proporciona una forma segura de administrar las cuentas y contraseñas de los usuarios (Django, 2022).

Django REST Framework

Django REST framework es un conjunto de herramientas potente y flexible para crear Interfaces de Programación de Aplicaciones (APIs por sus siglas en inglés) dando ganancias de usabilidad para los desarrolladores. Es personalizable por completo: solo usa las vistas regulares basadas en funciones si no necesita las funciones más potentes. Contiene amplia documentación. Es utilizado y confiado por empresas reconocidas internacionalmente, incluidas Mozilla, Red Hat, Heroku y Eventbrite. El marco REST requiere Python (3.6, 3.7, 3.8, 3.9, 3.10) y Django (2.2, 3.0, 3.1, 3.2, 4.0), aunque oficialmente la última versión del parche de cada serie de Python y Django. Las políticas de autenticación que incluyen paquetes para Autorización Abierta (OAuth1a por sus siglas en inglés Open Authorization) y OAuth2. Serialización que admite fuentes de Asignación Relacional de Objetos (ORM del inglés *Object Relational Mapping*) y no ORM (Django Rest Framework, 2022).

Bootstrap 4.6.0

Bootstrap es un framework de software libre para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript adicionales. Desde la versión 2.0 soporta diseños sensibles. Bootstrap es modular y consiste esencialmente en una serie de hojas de estilo LESS que implementan la variedad de componentes de la plataforma. El uso del lenguaje de hojas de estilo LESS permite el uso de variables, funciones y operadores, selectores anidados, así como clases mixin. Desde la versión 2.0, la configuración de Bootstrap también tiene una opción especial de "Personalizar" en la documentación. Los componentes de JavaScript para Bootstrap están basados en la librería jQuery de JavaScript. Los plugins se encuentran en la herramienta de plugin de jQuery. Proveen elementos adicionales de interfaz de usuario como diálogos, tooltips y carruseles.

1.6.7 Entornos de desarrollo integrado

Capítulo 1: Fundamentación teórica del módulo de administración remota del servicio telemático DHCP

Visual Studio

Visual Studio es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C# y Visual C++ utilizan todos los mismos entornos de desarrollo integrado (IDE), que habilita el uso compartido de herramientas y facilita la creación de soluciones en varios lenguajes. Así mismo, dichos lenguajes utilizan las funciones de NET Framework, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios Web XML

1.6.8 Librerías

React es una biblioteca de JavaScript para construir interfaces de usuario. Permite crear interfaces de usuario interactivas de forma sencilla. Diseña vistas simples para cada estado en tu aplicación, y se encarga de actualizar y renderizar de manera eficiente los componentes correctos cuando los datos cambien. Crea componentes encapsulados que manejen su propio estado, y conviértelos en interfaces de usuario complejas. Permite renderizar desde el servidor usando Node, así como potencializar aplicaciones móviles usando React Native (React, 2022)

Paramiko es una biblioteca de Python que se conecta de forma remota a otra computadora usando SSHv2 (*Secure Shell 2*), una versión avanzada de SSH. Es un protocolo de red para realizar una conexión segura entre dos ordenadores. Proporciona instalaciones tanto para el cliente como para el servidor. Podemos transferir archivos de forma remota de un ordenador a otro mediante SFTP (Protocolo de transferencia de archivos seguros) y SCP (Protocolo de copia segura) (Delftstack, 2022).

1.6.9 Protocolo de seguridad

SSH o Secure Shell, es un protocolo de administración remota que les permite a los usuarios controlar y modificar sus servidores remotos a través de Internet a través de un mecanismo de autenticación.

Proporciona un mecanismo para autenticar un usuario remoto, transferir entradas desde el cliente al host y retransmitir la salida de vuelta al cliente. El servicio se creó como un reemplazo seguro para el Telnet sin cifrar y utiliza técnicas criptográficas para garantizar que todas las comunicaciones hacia y desde el servidor remoto sucedan de manera encriptada (Deyimar, 2022).

1.7 Conclusiones parciales del capítulo

A modo de cierre los métodos teóricos mostraron las relaciones esenciales del objeto de investigación que no son observables y en consecuencia una interpretación conceptual de los datos empíricos encontrados. Los conceptos fundamentales dieron

Capítulo 1: Fundamentación teórica del módulo de administración remota del servicio telemático DHCP

un horizonte más exacto al objeto de estudio de la investigación. En conclusión, la búsqueda de información relacionada con el funcionamiento del servicio DHCP, su instalación y configuración propició las condiciones que debe tener la plataforma que se desarrollará. Como resultado del estudio de las herramientas que existen en la actualidad vinculadas al servicio DHCP, se demostró la necesidad de crear una plataforma que permita aplicarse a GNU/Linux Nova Servidor según las necesidades actuales de las empresas en el país. Como cierre, la metodología de software empleada en su variación para la UCI guía al proyecto de manera metodológica y didáctica. Como consecuencia de lo expuesto se definieron las herramientas y lenguaje para el modelo de la propuesta de solución que garanticen la seguridad, sostenibilidad y escalabilidad de la plataforma.

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

En el siguiente capítulo se hace descripción de la propuesta de solución de un módulo que garantice la administración remota del servicio DHCP a través de la plataforma Nova-ARST. Se exponen los requisitos funcionales y no funcionales del módulo, las técnicas con la que fueron identificados, la descripción de los mismos a través de historia de usuario y así como los prototipos de interfaz de usuario para cada caso de uso crítico.

2.1 Propuesta del módulo a desarrollar

Después de realizar un estudio de las principales herramientas vinculadas al servicio DHCP tanto en Cuba como en el Mundo, se busca desarrollar un módulo de automatización y reducción de los pasos necesarios para la configuración inicial y recurrente, incluyendo la creación de subredes, concesiones y vistas.

Los administradores de red con poca experiencia podrán configurar los servicios de DHCP en menos tiempo que con los procesos manuales. Por ejemplo, se tarda un par de minutos en configurar un DHCP de conmutación por error con una herramienta que integre DNS (*Domain Name System*), DHCP e IPAM (Administración de direcciones IP), mientras que puede llevar horas escribir los archivos de configuración adecuados.

Con una interfaz de usuario entre el administrador y el servidor, la edición de datos puede realizarse mediante pasos claros y lógicos, con lo que estas tareas se vuelven más intuitivas. La validación permite delegar con seguridad la administración en miembros menos experimentados del equipo. Cuando audita sus direcciones IP, quiere disponer de una imagen completa de la red Información precisa sobre qué subred está brindando servicios de DHCP. Los datos actuales e históricos sobre concesiones: qué IP se han concedido a qué clientes y cómo se utilizan. Permite conocer qué volumen de su espacio de direcciones está asignado actualmente a los clientes.

Con una plataforma dedicada a estos procesos, se puede obtener una visión integral en tiempo real del uso de direcciones IP del servicio de DHCP, con información gráfica fácil de leer y con los detalles de configuración de cada dirección expuestos claramente. Esto le permite controlar eficazmente la asignación de direcciones IP y poner en marcha un plan para atender a los CPE (Customer Provided Equipment del español Equipo Proporcionado por el Cliente) que entran y salen de la red. Cuando se tienen varios servidores distribuidos, una plataforma dedicada a la atención de estos procesos le otorgan al administrador una visibilidad global de todo su servicio de DHCP desde una interfaz centralizada.

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

La principal prioridad es garantizar la máxima disponibilidad de este servicio de red vital. Un módulo que valide puede suprimir los errores de configuración que pondrían en peligro la estabilidad del servicio de DHCP. Si una configuración propuesta contiene conflictos o valores inaceptables, el módulo se niega a implementar los cambios y proporciona una descripción detallada de qué datos son válidos y por qué. Tener un plan de respaldo para el servicio de DHCP protege aún más la red ante fallos costosos. Puede facilitar mucho la configuración de pares de DHCP de conmutación por error. Con unos pocos clics, puede establecer procesos que sincronicen los datos entre servidores, y configurar los parámetros para su mecanismo de conmutación por error (Appliansys, 2001). Para ello se desarrollará la plataforma Nova-ARST (Administración Remota de Servicios Telemáticos) que encierra las opciones necesarias para dar solución a lo antes mencionado y de prioridad a la administración remota que garantice a las empresas cubanas gestionar de forma rápida y eficiente su red interna según las necesidades de las mismas.

2.2 Requisitos

Los requisitos de un proyecto de software son las funciones, características y restricciones que debe cumplir el producto final. En otras palabras, los requisitos definen qué debe hacer el software, cómo debe verse y las condiciones que deben cumplirse para que se considere exitoso (Visure Solutions, 2020).

2.2.1 Fuentes para la obtención de requisitos

Las fuentes de obtención de requisitos utilizadas fueron:

- ✓ Análisis de las herramientas existentes (Ver **Epígrafe 1.4**).
- ✓ Debates con los especialistas de CESOL.

2.2.2 Técnicas de identificación de requisitos

Se obtuvo los requisitos gracias a la realización de un análisis de varias documentaciones relacionadas con este tema, entre ellas tesis, herramientas ya desplegadas en años anteriores en la universidad de carácter similar para una mayor comprensión, así como el estudio de herramientas con años de mercado agrupándose funcionalidades que no deben faltar en el módulo.

Entrevista

Se realizó una entrevista con el cliente para obtener información y convertirla de un lenguaje natural a uno más técnico para facilitar su desarrollo, consiste en reuniones analista-interesado en las cuales suceden preguntas y respuestas para extraer el dominio de la plataforma. La forma en que se realizó esta técnica fue: el autor de la

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

investigación se entrevistó con el cliente y especialistas de CESOL, a partir de la entrevista realizada se detectaron 9 requisitos funcionales y 7 requisitos no funcionales como se aprecia en la Tabla 2 y 3.

Mesas de trabajo

Se hicieron mesas de trabajo entre tutores y desarrollador de la plataforma para debatir con la técnica de **tormenta de ideas** logrando el aporte de ideas e iniciativas sobre el tema.

Por último, se obtuvo la historia de usuarios para una mayor comprensión por parte de los clientes, a la hora de hacer uso de la plataforma.

2.2.3 Descripción del módulo DHCP para la plataforma Nova-ARST

Con la entrevista realizada al ingeniero Lexys Manuel Días Alonso se concluye que el módulo del servicio telemático DHCP para la plataforma Nova-ARST está enfocado en la administración remota de servicios telemáticos que desarrollará el Centro de Software Libre (CESOL) para apoyar el proceso de migración hacia código abierto en las instituciones cubanas. Este módulo será desarrollado con el objetivo de que los administradores de servicios de red se apoyen de una plataforma que sustituya interfaces por las líneas de comandos introducidas de forma manual en la terminal del ordenador y que sean capaces de monitorear y configurar dichos servicios a través de la misma. Dentro de los servicios que se van a poder administrar están el DNS, PostgreSQL, servidores web Apache y Nginx, ProFTPd para un servicio en el que compartirá carpetas en la red, entre otros.

2.2.4 Especificación de requisitos de software

Una especificación de requisitos de software (ERS) proporciona una descripción completa de un producto de software a desarrollar y mantiene a todos los involucrados en la misma página. Ofrece una descripción completa del comportamiento del sistema a desarrollar. Incluye un conjunto de casos de uso que describe todas las interacciones que tendrán los usuarios con el software. A continuación, se describen los casos de uso o requisitos funcionales, así como los requisitos no funcionales de la propuesta de solución.

Requisitos funcionales

Los requisitos funcionales (RF) son declaraciones de los servicios que prestará la plataforma, la forma en que reaccionará a determinados insumos. Solo se implementarán las entradas en la herramienta, su transporte y salidas, ya sea cálculos, manipulación de datos, proceso comercial, interacción del usuario o cualquier otra

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

funcionalidad específica que permita definir qué función debe realizar la herramienta (PESSMAN, 2002). La complejidad de los requisitos funcionales se obtuvo mediante el producto de trabajo Evaluación de requisitos, propuesto en el expediente de proyecto 5.0 para la actividad productiva de la universidad.

Tabla 2: Requisitos funcionales del sistema. Elaboración propia

N.º	Nombre del RF	Descripción	Complejidad	Prioridad
1	Establecer Conexión Remota	El módulo DHCP debe ser capaz de acceder de forma remota a un ordenador determinado que se encuentre en la red, se ingresan los datos necesarios para dicho acceso.	Alta	Alta
2	Actualizar PC	El módulo DHCP debe ser capaz de actualizar los paquetes necesarios o desactualizados del ordenador accedido de forma remota.	Media	Alta
3	Instalar únicamente servicio DHCP	El módulo debe ser capaz de efectuar la instalación solo del servicio DHCP, instalados anteriormente en ordenador que es accedido de forma.	Media	Alta
4	Instalar isc-dhcp-common y sus paquetes dependientes	El módulo DHCP debe ser capaz de instalar el paquete isc-dhcp-common el cual permite la configuración dinámica y estática de este servicio telemático, así como los paquetes dependientes para su correcto funcionamiento.	Media	Alta
5	Desinstalar únicamente servicio DHCP	El módulo DHCP debe ser capaz de desinstalar únicamente el servicio DHCP, instalados anteriormente en ordenador que es accedido de forma.	Media	Media
6	Desinstalar DHCP y sus paquetes dependientes	El módulo DHCP debe ser capaz de desinstalar este servicio telemático y sus paquetes dependientes, instalados anteriormente en	Media	Alta

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

		ordenador que es accedido de forma.		
7	Eliminar información de configuración del servicio DHCP	El módulo DHCP debe ser capaz de eliminar todas las configuraciones que se hayan realizado con el servicio DHCP, instalados anteriormente en ordenador que es accedido de forma, restableciendo el servicio por defecto.	Media	Alta
8	Eliminar información de configuración del servicio DHCP y todos los paquetes dependientes.	El módulo DHCP debe ser capaz de eliminar todas las configuraciones que se hayan realizado con el servicio DHCP sus paquetes dependientes instalados anteriormente en ordenador que es accedido de forma, restableciendo el servicio por defecto.	Media	Alta
9	Iniciar servicio DHCP	El módulo debe ser capaz de iniciar el servicio DHCP ya configurado en el ordenador que es accedido de forma remota.	Media	Alta
10	Detener servicio DHCP	El módulo debe ser capaz de detener el servicio DHCP ya configurado en el ordenador que es accedido de forma remota.	Media	Alta
11	Reiniciar servidor DHCP	El módulo debe ser capaz de reiniciar el servicio DHCP ya configurado en el ordenador que es accedido de forma remota.	Media	Alta
12	Editar Adaptador de Red	El módulo debe ser capaz de editar el adaptador de red por el cual se conectará físicamente al cableado de red el ordenador al que fue accedido de forma remota.	Alta	Alta
13	Reservar dirección IP	El módulo debe ser capaz de reservar direcciones IP's necesarias para el control de		

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

		dispositivos específicos, el administrador introduce los datos necesarios para configurar esta opción del servicio DHCP.		
14	Configuración dinámica del servicio DHCP.	El módulo DHCP debe permitir la configuración dinámica de este servicio telemático, introduciendo los datos necesarios para su correcto funcionamiento.	Alta	Alta

Requisitos no funcionales

Se definen los requisitos no funcionales (RNF) del sistema, haciendo referencia a las cualidades, restricciones y características del software. Se caracterizan por ser específicos, cuantificables y verificables. Se refiere a todos los requisitos que no describen información a guardar, ni funcionar a realizar, sino características de funcionamiento. Suelen ser denominados como atributos de calidad del sistema (Pressman, 2002).

Tabla 3: Requisitos no funcionales del sistema. Elaboración propia

Número	Clasificación	RNF
1	Restricciones del diseño de implementación	Se utiliza el lenguaje de programación <i>Python 3.8</i> , <i>Django 2.2</i> , <i>Bootstrap 4.6.0</i> , <i>HTML5</i> , <i>CSS</i> , <i>JS</i> y <i>framework Django REST</i> .
2	Restricciones diseño	La interfaz visual debe mantener el estilo de la Plataforma de administración remota Nova-ARST.
3	Interoperabilidad	La plataforma para la administración remota de servidores DHCP debe ser compatible con la distribución cubana de GNU/Linux Nova Servidor 8.0.
4	Usabilidad	La plataforma debe adaptarse a las estrategias marcarias de la entidad en la que se haga uso o en la misma entidad en que se desarrolló.
5	Portabilidad	Se puede acceder a la plataforma desde navegadores web que soporten HTML5, CSS3 y JavaScript como Chrome o Firefox.
6	Confiabilidad	La plataforma debe ser capaz de salvar la información de forma automática en caso de caer el fluido eléctrico, fallo del ordenador en que se use o

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

		problemas funcionales de la misma.
--	--	------------------------------------

2.2.5 Descripción de requisitos de software

Se describe los requisitos funcionales de la propuesta de solución a través de las historias de usuario, para acercar al cliente y el desarrollador a un debate de cara a obtener el valor deseado y darle una explicación general e informal de las funciones de la plataforma desde la perspectiva del usuario final. A continuación, se muestran las historias de usuario correspondiente a los RF1 y RF3 de prioridad Alta, las restantes historias de usuario se mostradas a partir del Anexo2.

Tabla 4: Historia de usuario Instalar servicio DHCP. Elaboración propia


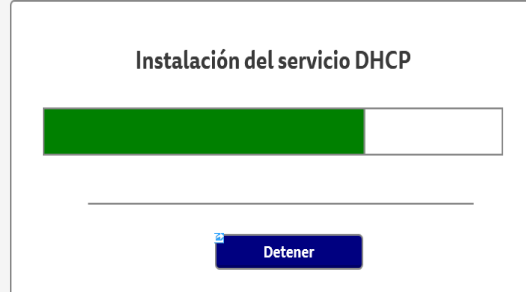
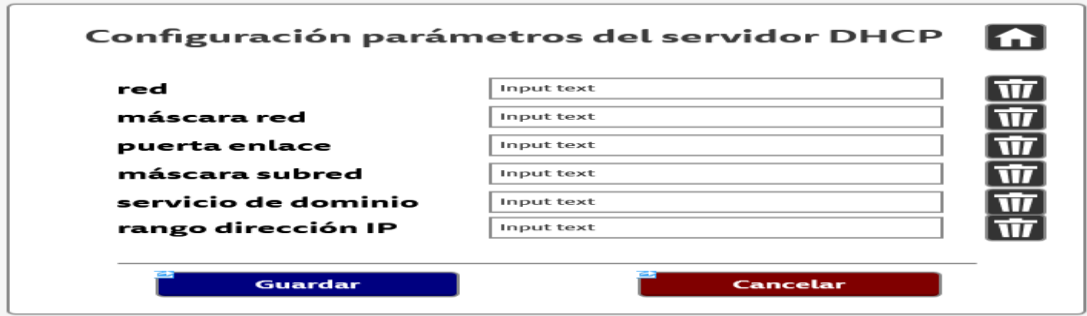
Historia de usuario	
Numero: HU_1	Nombre: Instalar servidor DHCP
Prioridad: Alta	Iteración Asignada: 1
Programador: Reynier Fernández Rodríguez	Tiempo Estimado: 1 semana
Riesgo en Desarrollo: N/A	Tiempo Real: 1 semana
Descripción: El sistema comienza el proceso de instalación del servicio DHCP de forma automática una vez el administrador pincha esta opción	
Observaciones: El administrador debe acceder al módulo DHCP de la plataforma Nova-ARST y en caso de no estar instalado el servicio le aparecerá una pantalla de bienvenida priorizando la instalación del servicio DHCP, una vez pinche iniciar le aparecerá una ventana de instalación <ul style="list-style-type: none"> ✓ Verificar que la máquina en la que se encuentra tenga conexión a <i>internet</i> ✓ Confirmar instalación 	
Prototipo:	
	

Tabla 5: Historia de usuario Configurar parámetros del servidor DHCP. Elaboración propia

Historia de usuario	
Numero: HU_3	Nombre: Configurar parámetros del servidor DHCP
Prioridad: Alta	Iteración Asignada: 1

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

Programador: Reynier Fernández Rodríguez	Tiempo Estimado: 1 semana
Riesgo en Desarrollo: N/A	Tiempo Real: 2 semana
Descripción: La plataforma permite al administrador configurar los parámetros necesarios en un servidor DHCP.	
Observaciones: El administrador debe acceder al módulo DHCP de la plataforma Nova-ARST y buscar la opción de Configuración de parámetros del servidor DHCP, para ello es necesario editar los siguientes parámetros: <ul style="list-style-type: none">✓ Red✓ Máscara red✓ Puerta de enlace✓ Máscara subred✓ Servicio de dominio✓ Rango dirección IP	
Prototipo: 	

2.2.6 Validación de requisitos

La validación de requerimiento es un proceso continuo en el proyecto de desarrollo de software con el fin de asegurar que los requisitos sean representaciones exactas de las necesidades expectativas de los usuarios. Esta actividad contribuye a mejorar la calidad de los requisitos, reducir costos, tiempo y riesgo en el desarrollo de la plataforma. A continuación, se describen las técnicas de validación de requisitos utilizadas en la propuesta de solución:

Prototipo de interfaz de usuario: la interfaz de usuario (UI en inglés) es “el diseño de cualquier aplicación web que utiliza un usuario a través de la interacción con sistemas en internet”. La definición formal de la palabra prototipo habla de aquel “ejemplar original o primer molde” de un artefacto a fabricarse. En sentido general, también podemos decir que se trata de “una implementación parcial pero concreta de un sistema, que se crea para explorar aspectos diversos durante el desarrollo del mismo”. Es una simulación

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

interactiva y real de cómo funcionará una aplicación web que permite ver los flujos de entradas y salidas, ya sea de una sola sección o de un producto digital completo.

Los prototipos de interfaz de usuario realizados de acuerdo a los requisitos obtenidos, le permiten al autor ordenar las ideas, explorar diferentes caminos de concepto o diseño, y detectar posibles problemas o carencias antes de empezar la fase de programación de la plataforma. Además de probar, analizar, ajustar y perfeccionar los elementos con los que van a interactuar los usuarios de la plataforma.

2.3 Análisis y diseño

El análisis y diseño de sistemas tiene como objetivo planificar el desarrollo de los sistemas de información mediante la comprensión y la especificación en detalle de lo que debe hacer un sistema y cómo deben implementarse los diferentes componentes del mismo para trabajar conjuntamente (Teorema, 2020).

2.3.1 Diseño arquitectónico

El diseño de una arquitectura de software utiliza los conocimientos de programación para planear el diseño general del software de modo que puedan agregarse detalles más adelante, permite a los equipos de software delimitar el panorama general y comenzar a elaborar un prototipo. Si los desarrolladores de software siguen los consejos de diseño y las prácticas recomendadas de arquitectura de software, pueden analizar detalladamente las características de su software y decidir cómo diseñar la arquitectura de este (Lucidchart, 2022).

La arquitectura que se utilizará para el desarrollo del módulo en la plataforma de administración remota del servicio DHCP será de N-Capas, en específico 3 Capas, la cual permite dividir la plataforma en capas lógicas y niveles físicos, se separa responsabilidades a través de módulos y se administran las dependencias entre ellas. A continuación, se muestra en la Figura 1 la descripción de la arquitectura aplicada.

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

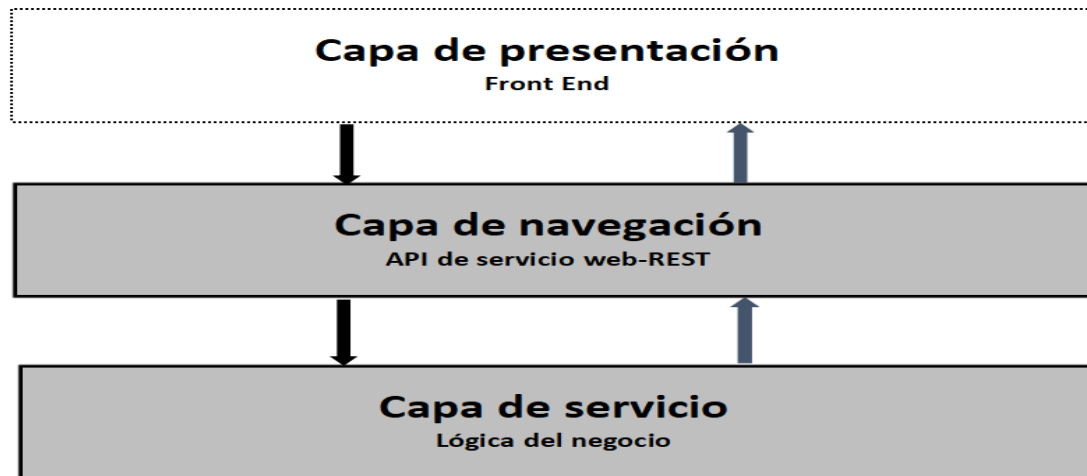


Figura 1: Arquitectura de la plataforma para la administración remota del servicio DHCP.

Elaboración propia.

Las capas son una forma de separar responsabilidades y administrar dependencias. Cada capa tiene una responsabilidad específica. Una capa superior puede utilizar los servicios de una capa inferior, pero no al revés.

- ✓ Capa de presentación: es la que ve el usuario (también se le denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de rutas.
- ✓ Capa de navegación: existen cuatro tipos de API (del español Interfaz de Programación de Aplicaciones) de servicios web habituales entre los desarrolladores, en esta plataforma se utilizará la de tipo REST (*Representational State Transfer* del español Transferencia de Estado Representacional) que permite en la capa de navegación el intercambio de información entre un servicio web (*software* que da acceso a un servicio concreto a través de una URL) y una aplicación. Normalmente ese intercambio se produce a través de peticiones HTTP o HTTPS (la versión cifrada del protocolo HTTP).
- ✓ Capa de servicio: los datos son transformados o calculados en la capa de servicio. Se llevan a cabo esos cálculos antes de pasarlos al servidor, la petición del usuario se ejecuta dentro de una promesa de datos que se resuelven con la respuesta del servidor y retorna a la capa de navegación, seguido la capa de presentación devolviendo los datos solicitados al usuario.

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

2.3.2 Persistencia de datos

El servicio DHCP se configura a través de ficheros que contienen las estructuras necesarias en el que el servidor reconozca de forma exacta la información para su correcto funcionamiento, las localizaciones de estos ficheros se encuentran en direcciones la cual no se tiene permiso de escritura, por tanto, no se pueden entrar a estos archivos y modificar de forma manual los mismos, solo se puede realizar estas operaciones en el mismo a través de consola, es por eso que se desarrollaron métodos necesarios para el acceso remoto de estos archivos en cualquier ordenador que se encuentre en la red, así como su copia al ordenador donde se encuentre la plataforma Nova-ARST en una dirección que si permita la escritura en de cualquier archivo, luego este archivo se rellena con los datos entrados a través de los formularios por el administrador de la plataforma y se colocan de forma organizada la información dentro del archivo sin alterar su correcto funcionamiento.

2.4 Diagrama de paquetes

Se diseñó el diagrama de paquete para mostrar la organización y disposición de diversos elementos del modelo en forma de paquetes. Los paquetes son una agrupación de elementos UML relacionados, como diagramas, documentos, clases o, incluso, otros paquetes. Cada elemento está anidado dentro de un paquete, que se representa como una carpeta de archivos dentro del diagrama, y que luego se organiza de forma jerárquica dentro del diagrama. Se usa para proporcionar una organización visual de la arquitectura en capas de la plataforma Nova-ARST. En el diagrama de la Figura 2 se muestra la relación entre las capas de presentación, navegación y servicio, así como las dependencias y elementos que abarcan.

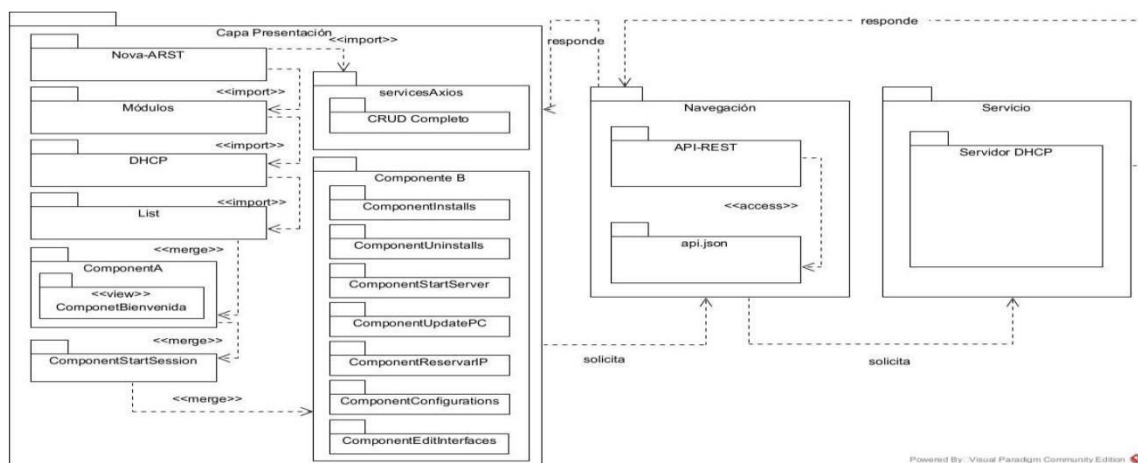


Figura 2: Diagrama de paquetes del diseño arquitectónico. Elaboración propia.

2.5 Patrón de arquitectura React Composition.

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

La composición es el acto de combinar partes o elementos para formar un todo. Los componentes son los “*building blocks*” básicos de la interfaz de usuario en las aplicaciones React, al igual que las funciones puras son los componentes básicos de la composición de funciones. *React Composition* es un patrón de desarrollo basado en el modelo de componentes original de React en el que construimos componentes a partir de otros componentes utilizando accesorios definidos explícitos o los accesorios secundarios implícitos. En términos de refactorización, la composición de React es un patrón que se puede usar para dividir un componente complejo en componentes más pequeños y luego componer esos componentes más pequeños para estructurar y completar su aplicación.

2.6 Modelo de diseño

El modelo de diseño es planteado como un modelo de objetos que describe la realización física de los casos de usos, centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar, constituyendo una entrada principal en la actividad de implementación (López, 2015).

2.6.1 Diagrama de clases

Los diagramas de clase (DC) describen los tipos de objetos de un sistema, así como los distintos tipos de relaciones que pueden existir entre ellos. Es una herramienta para comunicar el diseño de un programa orientado a objetos, permitiendo modelar las relaciones entre las entidades (Francisco José, Carlos Pard, 2022). A continuación, se muestra en la Figura 3 y 4 el diseño del DC con estereotipo web para describir de la propuesta de solución con un nivel más técnico para el programador de la plataforma, el resto de los diseños se quedaron plasmados en los anexos:

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

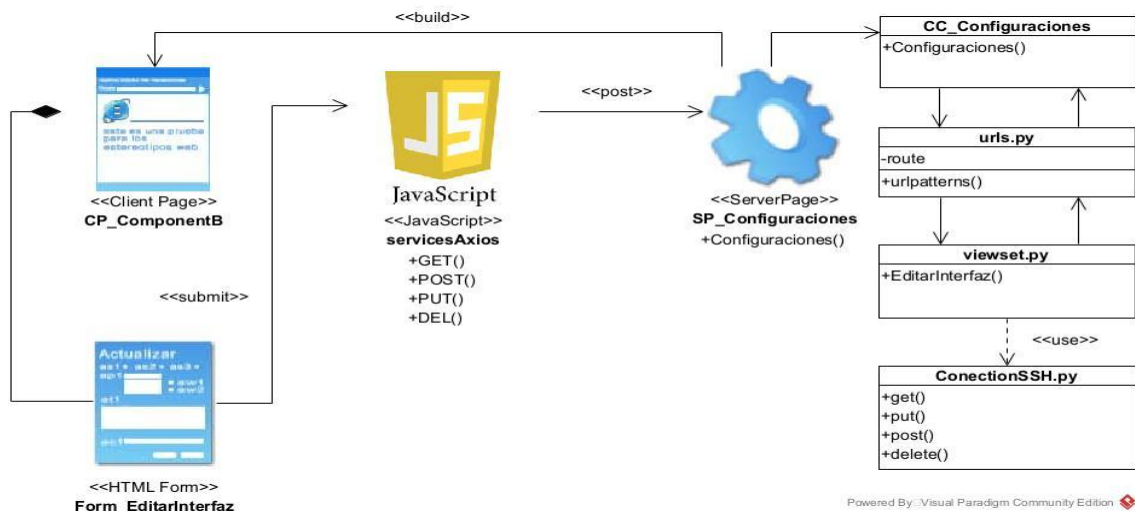


Figura 3: Diagrama de clases del diseño con estereotipos web de la HU Editar Interfaz DHCP”.

Elaboración Propia

<<Client Page ComponentB>>: Página web con formato XHTML. Muestra al usuario la información de la capa de presentación, las opciones o requisitos funcionales del módulo DHCP.

<<Form_EditarInterfaz>>: Colección de datos para la edición de la interfaz para el adaptador de red, están contenidos en la página cliente. Sus atributos son los elementos de entrada del formulario. No tienen operaciones y se comunican con las páginas servidoras mediante submit (envió de datos).

<<JavaScript>>: Medio de comunicación entre la capa de presentación y API, abarca los métodos get, put, post y del lado del front end.

<<Server Page Configuraciones>>: Representa la clase que tiene código que se ejecuta en el servidor, la cual se encarga de construir (build), generar el resultado HTML y realizar peticiones al back end.

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

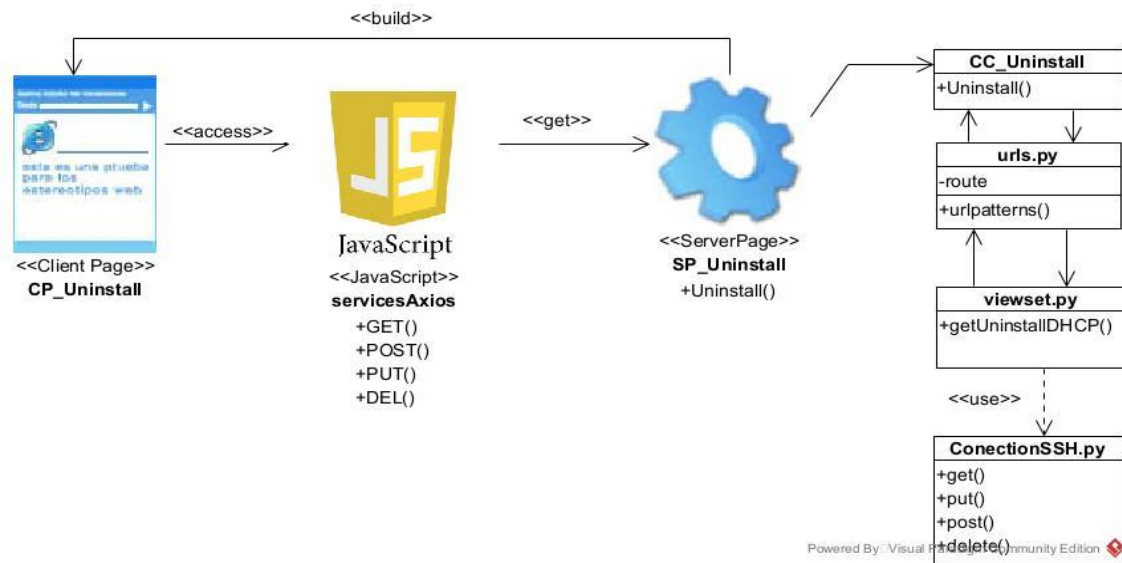


Figura 4: Diagrama de clases del diseño con estereotipos web de la HU "Desinstalar solo el servicio DHCP". Elaboración Propia

<<Client Page Uninstall>>: Presenta al cliente la información visual de las desinstalaciones relacionadas con el servicio DHCP.

<<build>>: Es una relación direccional, donde una página servidora construye un o más páginas cliente.

<<submit>>: Envía los valores de un formulario a una página servidora.

2.6.2 Patrones de diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Contribuyen a reutilizar diseño gráfico identificando aspectos claves de la estructura de un diseño que puede ser aplicado en una gran cantidad de situaciones (PRESMAN, 2002). A continuación, se describen los patrones aplicados para la implementación de la propuesta de solución:

Patrones GRASP

General Responsibility Assignment Patterns (GRASP), son una serie de patrones que describen los principios fundamentales de la asignación de responsabilidades a objetos y son considerados una serie de buenas prácticas en el diseño de software (LARMAN, 1999).

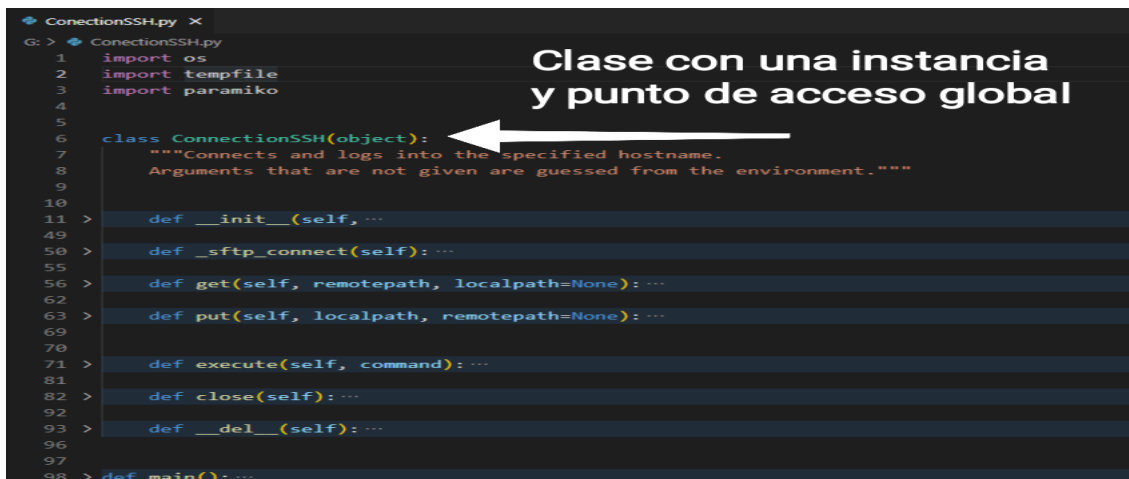
Experto

Experto es un patrón que suele utilizarse en el diseño orientado a objetos. Este patrón sugiere asignar responsabilidad al objeto que posea la información necesaria para

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

desempeñarla. Con la utilización se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. El comportamiento se distribuye entre las clases que cuentan con la información requerida, alentando con ello definiciones de clases sencillas y más cohesivas que son más fáciles de comprender y mantener.

Este patrón se evidencia en el componente Componente B que tiene acceso a todas las entidades necesarias y por ello a la información, por lo cual se le asigna la responsabilidad de modelar todos los atributos de configuración como se muestra en la Figura 5.



```
ConectionSSH.py X
G: > ConectionSSH.py
1 import os
2 import tempfile
3 import paramiko
4
5
6 class ConnectionSSH(object):
7     """Connects and logs into the specified hostname.
8     Arguments that are not given are guessed from the environment."""
9
10
11 > def __init__(self, ...
49
50 > def _sftp_connect(self): ...
55
56 > def get(self, remotepath, localpath=None): ...
62
63 > def put(self, localpath, remotepath=None): ...
69
70
71 > def execute(self, command): ...
81
82 > def close(self): ...
92
93 > def __del__(self): ...
96
97
98 > def main(): ...
```

Clase con una instancia y punto de acceso global

Figura 5: Clase experta en información. Elaboración propia.

Creador

Este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos, su propósito es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. Brinda soporte a un bajo acoplamiento, lo que supone menos dependencias respecto al mantenimiento y mejores oportunidades de reutilización como se muestra en la Figura 6

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

```
6  const ComponenteB = ({titulo, texto, carrucell1, boton1, boton2}) => {
7  return (
8  <div className={styles.contenidoAa}>
9    <h4>
10   <Card.Header className={styles.cartaCabeza}>{titulo}</Card.
11   Header>
12   </h4>
13   <div >
14     { <img className={styles.imagen} src={carrucell1}
15     alt="imagenes" />}
16   </div>
17   <span>
18     <Card.Text>{texto}</Card.Text>
19   </span>
20   <div>
21     | {boton1}
22   </div>
23   <div>
24     | {boton2}
25   </div>
26   <span>
27     <Card.Footer></Card.Footer>
28   </span>
29
30   </div>
31   </div>
32   </div>
33   );
```

línea 6: Creación de los objetos pertenecientes al ComponenteB

Figura 6: Ejemplo código de creación de objeto en la plataforma. Elaboración propia.

Patrones GOF

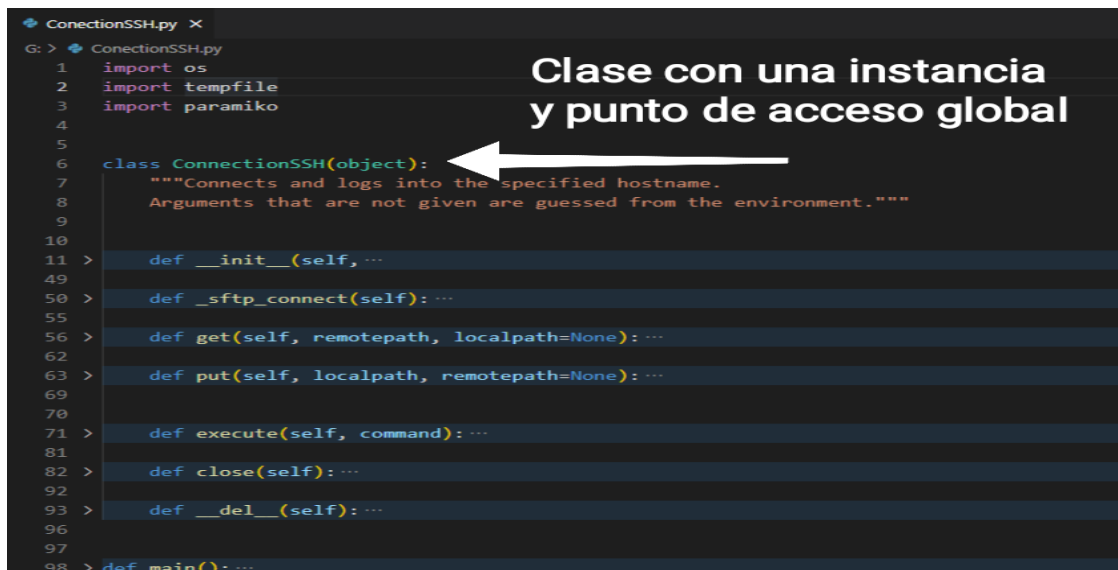
Dentro de los patrones clásicos tenemos los GoF (Gang of Four) basado en el libro Design Patterns (Diseños de Patrones). Las ventajas del uso de patrones son evidentes: Conforman un amplio catálogo de problemas y soluciones. Estandarizan la resolución de determinados problemas. Condensan y simplifican el aprendizaje de las buenas prácticas. Se contemplan 3 tipos de patrones:

- ✓ Patrones de creación tratan de la inicialización y configuración de clases y objetos.
- ✓ Patrones estructurales tratan de desacoplar interfaz e implementación de clases y objetos.
- ✓ Patrones de comportamiento tratan de las interacciones dinámicas entre sociedades de clases y objetos.

Singleton

Esta dentro del patrón de creación mencionado anteriormente, garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella. Se muestra un ejemplo de la aplicación de este patrón en la Figura 7.

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST



```
ConectionSSH.py X
G: > ConectionSSH.py
1 import os
2 import tempfile
3 import paramiko
4
5
6 class ConnectionSSH(object):
7     """Connects and logs into the specified hostname.
8     Arguments that are not given are guessed from the environment."""
9
10
11 > def __init__(self, ...
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49 > def _sftp_connect(self): ...
50
51
52
53
54
55 > def get(self, remotepath, localpath=None): ...
56
57
58
59
60
61
62 > def put(self, localpath, remotepath=None): ...
63
64
65
66
67
68
69
70
71 > def execute(self, command): ...
72
73
74
75
76
77
78
79
80
81 > def close(self): ...
82
83
84
85
86
87
88
89
90
91
92 > def __del__(self): ...
93
94
95
96
97
98 > def main(): ...
```

Figura 7: Patrón Singleton de la propuesta de solución. Elaboración propia.

2.7 Conclusiones parciales

Se dio respuesta a la solicitud de propuesta que hace el cliente. Mediante la entrevista realizada al cliente se definieron 14 requisitos funcionales y 6 no funcionales, dejando claro los de mayor prioridad y complejidad a la hora de su desarrollo. Tal y como se evidencia en los epígrafes anteriores, la historia de usuario, describieron los requisitos funcionales del sistema, dejando claro el tiempo aproximado de su desarrollo, posibles prototipos de interfaces que definirán la vista de la plataforma y observaciones a tener en cuenta a la hora de su ejecución. Se diseñaron a través de casos de prueba los posibles escenarios a la hora de ejecutar los requisitos de la plataforma. Se concluye que el análisis y diseño definió el diseño arquitectónico de la plataforma, ubicando en tres capas el módulo que será desarrollado y por ende garantizando que sean mejorados en un futuro sin afectar el resto de los mismos. Tras el diseño de los diagramas de clases con estereotipos web, se mostraron modelos que acercan la propuesta de solución a un nivel técnico para mejor entendimiento por parte del programador que desarrollará la plataforma. Para finalizar, el empleo de patrones de diseño obtuvo una solución de software con poca dependencia entre clases, flexible al mantenimiento y a la aceptación de cambios a corto y mediano plazo.

Capítulo 3. Implementación y pruebas al módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST

Capítulo 3. Implementación y pruebas al módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST

En este capítulo se muestran los diferentes artefactos que se utilizan para la implementación y pruebas de la plataforma, así como los estándares de codificación que debe seguir el equipo de desarrollo para un mejor entendimiento y organización del código. De acuerdo a la metodología que se utiliza, se especifican, de los tipos de pruebas que esta plantea, los que serán empleados para darle validez a los requisitos funcionales y garantizar el óptimo funcionamiento de la aplicación.

3.1 Diagrama de despliegue

Los diagramas de despliegue se utilizan para visualizar los procesadores/nodos/dispositivos de hardware de un sistema, los enlaces de comunicación entre ellos y la colocación de los archivos de software en ese hardware. Como propósito principal captura la configuración de los elementos de procesamiento, así como las conexiones entre estos elementos en el sistema. El modelo consiste en uno o más nodos, dispositivos y conectores, entre otros. Este modelo también permite mapear procesos dentro de estos elementos de procesamiento, permitiendo la distribución del comportamiento a través de los nodos que son representados. A continuación, se muestra en la Figura 8 el diagrama de despliegue modelado para la herramienta a desarrollar:

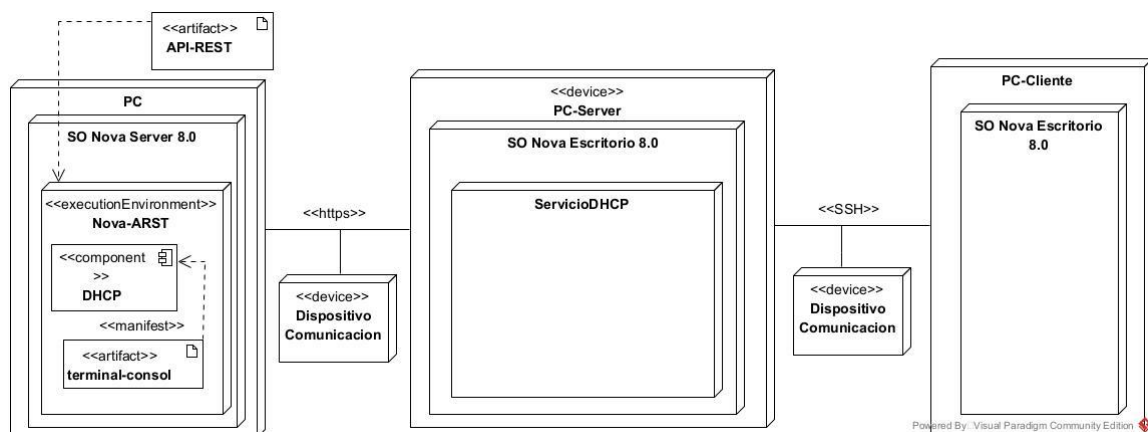


Figura 8: Diagrama de despliegue de la plataforma Nova-ARST

En la figura anterior se representa el diagrama de despliegue de la plataforma Nova-ARST, en el mismo se ilustran las relaciones hardware-software necesarias para un correcto funcionamiento de la plataforma en un ambiente de trabajo real y óptimo. La plataforma podrá ser gestionada por un administrador de redes y contará con un módulo del servicio telemático DHCP que le permita organizar a los clientes que se encuentren en la red de una empresa determinada. La comunicación cliente-servidor será a través

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

del protocolo TCP/IP y SSH, este último garantiza la seguridad de la información enviada entre ambas partes.

3.2 Modelo de implementación

Según Pressman el modelo de implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes se encuentran datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos.

3.2.1 Diagrama de componentes

Los diagramas de componentes se utilizan para visualizar la organización de los componentes de un sistema y las relaciones de dependencia entre ellos. Proporcionan una visión de alto nivel de los componentes de un sistema. Se utilizan en el desarrollo basado en componentes para describir sistemas con arquitectura orientada a servicios. Mostrar la estructura del propio código. Se puede utilizar para centrarse en la relación entre los componentes mientras se ocultan los detalles de las especificaciones. Ayudar a comunicar y explicar a los interesados las funciones del sistema que se está construyendo (CinergixPty.Ltd.,2008-2022). La Figura 9 muestra el diagrama de componentes correspondiente al módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST.

Capítulo 3. Implementación y pruebas al módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST

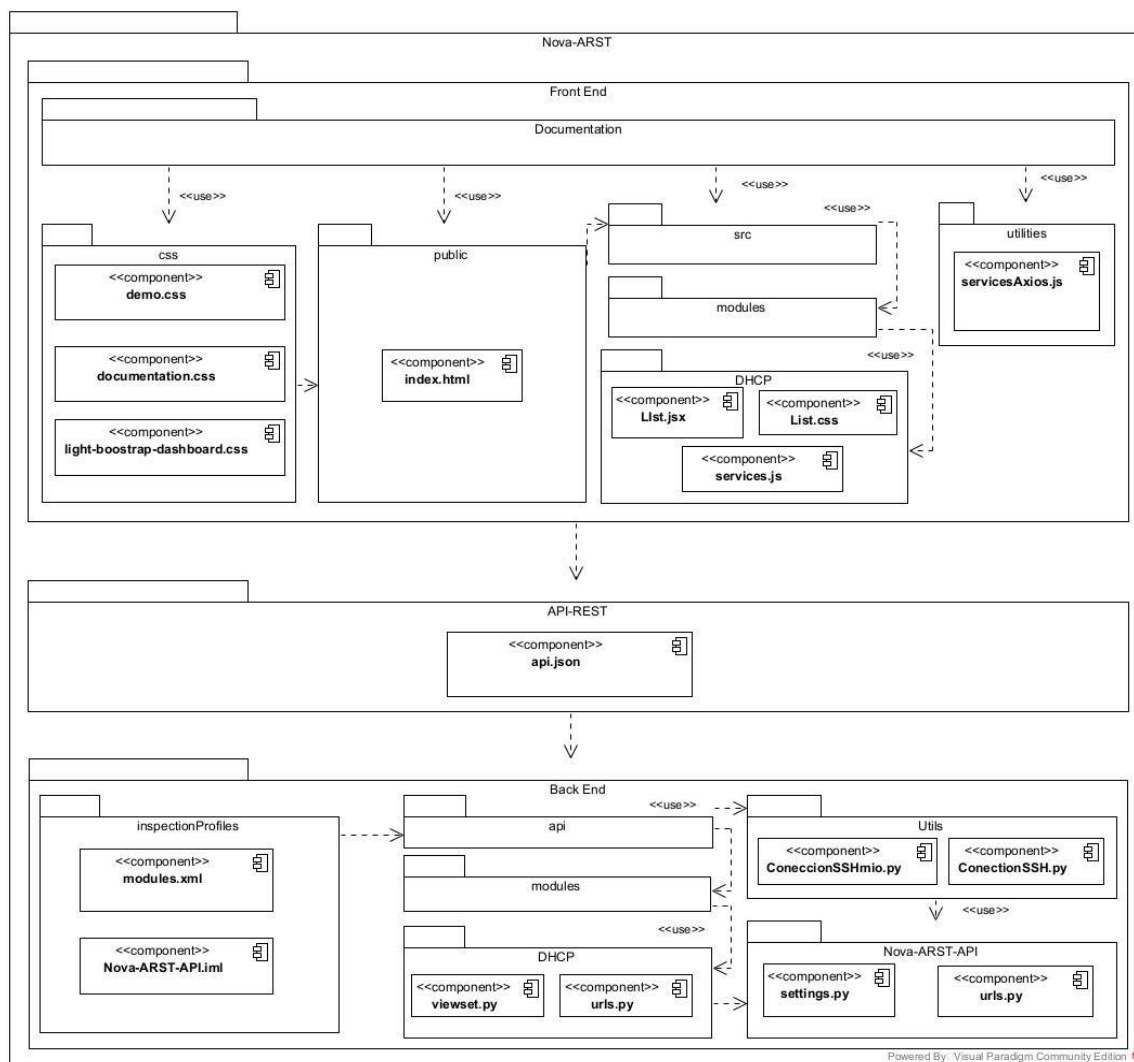


Figura 9: Diagrama de componentes del módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST. Elaboración Propia.

Descripción de los componentes del sistema

Front End: paquete que trabaja la interfaz de usuario desde el punto de vista del código, para que la interacción con la plataforma sea posible. Por lo tanto, se encarga de la parte visual de la web.

API-REST: paquete de definiciones y protocolos que se utilizan para diseñar e integrar el software de las aplicaciones. Suele considerarse con el contrato entre el proveedor de información y el usuario, donde se establece el contenido que se necesita por parte del consumidor (la llamada) y el que requiere el productor (la respuesta).

Back End: paquete que agrupa las funciones necesarias para acceder a la información que se solicita, a través de la plataforma, para luego combinarla y devolverla al usuario final.

Capítulo 3. Implementación y pruebas al módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST

3.3 Estándar de codificación

El objetivo de los estándares de codificación de software es inculcar prácticas de programación probadas que conduzcan a un código seguro, confiable, comprobable y mantenible. Por lo general, esto significa evitar las prácticas de codificación inseguras conocidas o el código que puede causar un comportamiento impredecible. En el siguiente epígrafe se muestran los estándares de los lenguajes de programación de React JS y Python basado en la guía de estilo del código Python por Guido Van Rossum y Barry Warsaw usados en el *front end* y *back end* de la plataforma respectivamente (Programador clic, 2020-2022).

React JS

Especificaciones básicas

- ✓ Cada archivo contiene solo un componente de React, y el nombre debe ser el que indica el nombre.
- ✓ React Native y React deben usar la sintaxis ES6 tanto como sea posible.
- ✓ La consola se escribe durante la depuración y se elimina inmediatamente después de la depuración.

Convención de nomenclatura

Nombre del archivo: Los nombres de carpetas y archivos deben usar nombres de mayúsculas y minúsculas, como *HomeIndex.js*; Los componentes comunes se colocan en la carpeta *Component*. El archivo de entrada debe ser claro, generalmente se puede usar *index.js*. En React Native, si solo hay un extremo de la función android / ios, agregue. android /. ios al final del archivo, por ejemplo: *Keyboard.android.js* y *Keyboard.ios.js*. Evite nombres como 1, 2, 3, uno, dos y tres, y *nómbrelos* con funciones o módulos. Todas las llamadas a Android / iOS nativo se escriben en el mismo archivo .js. Si es uno versátil como *Dialog*, se puede colocar en la carpeta *BaseView*, de lo contrario, se puede colocar en la carpeta de negocios correspondiente a la Acción.

Propiedades

- ✓ Nombre de la propiedad: Las propiedades del componente utilizan la nomenclatura de la pequeña caja camel. En React, *className* reemplaza el atributo de clase y *htmlFor* reemplaza el atributo for. Cuando se utilizan estilos fuera de línea, el nombre del atributo debe tener preferiblemente la palabra clave *Style*. Por ejemplo: *flexStyle.js*

Capítulo 3. Implementación y pruebas al módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST

- ✓ Configuración de la propiedad: Asegúrese de establecer propiedades en el componente empaquetado (para la verificación de propTypes) y no cambie el valor de la propiedad externamente. Las propiedades suelen utilizar la sintaxis {... this.props}. (Es mejor agregar (... this.props) al comienzo de la propiedad, para evitar escribir menos propiedades y generar errores). Si el valor del atributo se establece repetidamente, este último sobrescribirá el valor anterior.
- ✓ Alineación de atributos: Se puede organizar en filas cuando hay menos atributos. Cuando hay muchos atributos, un atributo por línea y la etiqueta cerrada está en una línea separada
- ✓ Iteración en línea: La lógica de operación es simple y utiliza directamente la iteración en línea (Anexo 7).

Comentarios

- ✓ Los comentarios entre componentes deben incluirse con `{/ * contenido de comentario * /}` (Anexo 8).

Uso de comillas

- ✓ Los atributos HTML / JSX utilizan comillas dobles `"`;
- ✓ JS usa comillas simples `'`; (Anexo 9).

HTML condicional

- ✓ La salida corta es un operador ternario directamente en línea; (Anexo 5).
- ✓ Las estructuras más complejas pueden definir una variable terminada en Html en el método. `render ()`. (Anexo 10).

() Uso

- ✓ JSX multilínea usa `()` para envolver, y el modo multilínea se usa cuando los componentes están anidados; (Anexo 11).
- ✓ JSX de una línea omite `()`. (Anexo 12).

Etiqueta de cierre automático

- ✓ Todas las etiquetas en JSX deben estar cerradas;
- ✓ Los componentes sin elementos secundarios utilizan una sintaxis de cierre automático, con un espacio antes de la etiqueta de cierre automático `/`. (Anexo 9).

Organización del código interno del componente

- ✓ No use el prefijo de subrayado para nombrar los métodos del componente React; (Anexo 13).

Capítulo 3. Implementación y pruebas al módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST

- ✓ Organizar los métodos y atributos de los componentes en el orden de la organización del ciclo de vida;
- ✓ Deje una línea entre los métodos (atributos);
- ✓ El método. render () siempre está al final;

Método personalizado después del método React API, antes de. render (). (Anexo 14).

Python

Indentación □

Usa cuatro espacios por cada nivel de codificación.

Las líneas de continuación deben alinearse verticalmente con el carácter que se ha utilizado (paréntesis, llaves, corchetes). □

No se mezclarán tabuladores y espacios en la codificación. El método de indentación más popular en Python es con espacios. El segundo más popular con tabulaciones, sin mezclar unos con otros. Cualquier código indentado con una mezcla de espacios y tabulaciones debe ser convertido a espacios exclusivamente.

Máxima longitud de las líneas

Todas las líneas deben estar limitadas a un máximo de 79 caracteres. Dentro de paréntesis, corchetes o llaves se puede utilizar la continuación implícita para cortar las líneas largas. En cualquier circunstancia se puede utilizar el carácter “\” para cortar líneas largas.

Líneas en blanco

Utilizar la codificación UTF-8. □ Se pueden incluir caracteres que no correspondan a esta codificación utilizando “\x”, “\u”, “\U” para cadenas (strings).

Importación

Las importaciones deben estar en líneas separadas. □

Las importaciones siempre se colocan al comienzo del archivo, simplemente luego de cualquier comentario o documentación del módulo, y antes de globales y constantes.

Deben quedar agrupadas de la siguiente forma:

1. Importaciones de la librería estándar.
2. Importaciones terceras relacionadas.
3. Importaciones locales de la aplicación/librerías. □

Cada grupo de importaciones debe de estar separado por una línea en blanco.

Espacios en blanco en expresiones y sentencias

Evitar usar espacios en blanco en las siguientes situaciones:

1. Inmediatamente dentro de paréntesis, corchetes o llaves.

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

2. Inmediatamente antes de una coma, un punto y coma o dos puntos.

3. Inmediatamente antes de un corchete que empieza una indexación.

4. Más de un espacio alrededor de un operador de asignación u otro para alinearlos con otro

Deben rodearse de espacios en blanco los siguientes operadores:

1. Asignación (“=”).

2. Asignación de aumentación (“+=”, “-=”, “*=”, “/=”).

3. Comparación (“==”, “<”, “>”, “>=”, “<=”, “! =”, “<>”, “in”, “not in”, “is not”).

4. Expresiones lógicas.

Si se utilizan operadores con prioridad diferente se aconseja rodear con espacios a los operadores de menor prioridad.

No utilizar espacios alrededor del igual (=) cuando es utilizado para indicar un argumento de una función o un parámetro con un valor por defecto.

Comentarios

Los comentarios deben ser oraciones completas.

Si un comentario es una frase u oración su primera palabra debe comenzar con mayúscula a menos que sea un identificador que comience con minúscula.

Si un comentario es corto el punto final puede omitirse.

Los comentarios de una línea para aclaraciones del código aparecerán seguidos de los caracteres “//” en caso de código JavaScript mientras que en Python por el carácter “#” y deben ubicarse en la misma línea que se desea comentar.

Los comentarios de varias líneas para organización del código aparecerán dentro de los caracteres “/** ... */” en caso de código JavaScript, mientras que en Python se hará con los caracteres “...”.

Convecciones de nombramiento

Nunca se deben utilizar como simples caracteres para nombres de variables los caracteres ele minúscula “l”, o mayúscula “O”, ele mayúscula “L” ya que en algunas fuentes son indistinguibles de los números uno (1) y cero (0).

Los módulos deben tener un nombre corto y en minúscula.

Los nombres de clases deben utilizar la convención “CapWords” (palabras que comienzan con mayúscula).

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

Los nombres de las excepciones deben estar escrito también en la convención “CapWords” utilizando el sufijo “Error”.

Los nombres de las funciones deben estar escritos en minúscula separando las palabras con un guion bajo (_).

Las constantes deben quedar escritas con letras mayúsculas separando las palabras con un guion bajo (_).

La Figura 10 representa un ejemplo donde se muestra la aplicación de algunos estándares de codificación.

```
117 def crearuser(host,user,password):
118
119     comando = 'sudo -u postgres createuser --superuser --pwprompt Gabriel'
120     client = paramiko.SSHClient()
121     client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
122     client.connect(host, username=user, password=password)
123
124     stdin, stdout, stderr = client.exec_command(comando, get_pty=True)
125
126     stdin.write('ijj\n')
127     stdin.flush()
128
129     result = ''
130
131
132     for line in stdout.read().splitlines():
133         result += f'{line} '
134         print(result)
135
136     e = stderr.read().decode()
137     data = {
138         'salida': result,
139         'error': e
140     }
141     print(data)
142
143     return data
144
```



Figura 10: Aplicación de los estándares de codificación. Elaboración propia.

Luego de realizar el modelado de implementación y definido los estándares de codificación utilizados para comprender todos los aspectos a tener en cuenta en la generación de código se define la estrategia para lograr una mejor organización y se realizan las pruebas de software.

3.4 Pruebas de software

Las pruebas de software son un conjunto de herramientas, técnicas y métodos que evalúan la excelencia, el desempeño de un software, involucra las operaciones del sistema bajo condiciones controladas y evalúa los resultados. Las técnicas para encontrar problemas en un programa son variadas y van desde el uso del ingenio por parte del personal de prueba hasta herramientas automatizadas que ayudan a aliviar el peso y el costo de tiempo de esta actividad (PRESSMAN, 2002). Una vez se

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

implementa el código del sistema es necesario probarlo para encontrar y erradicar la mayor cantidad de errores antes de entregarlo al cliente. A continuación, se describen los tipos de pruebas de software aplicadas, así como los métodos y técnicas empleada para la evaluación del módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST.

3.5 Aplicación de las pruebas de software

En el epígrafe se describen las pruebas de software realizadas en las disciplinas de Pruebas internas y Pruebas de aceptación propuestas en la metodología de desarrollo de software AUP-UCI.

3.5.1 Pruebas unitarias

Las pruebas unitarias son de muy bajo nivel y se realizan cerca de la fuente de la aplicación. Consisten en probar métodos y funciones individuales de las clases, componentes o módulos que usa tu software. En general, las pruebas unitarias son bastante baratas de automatizar y se pueden ejecutar rápidamente mediante un servidor de integración continua. Las pruebas unitarias se centran en probar cada componente de código de un software de forma individual para asegurar que funcione de manera apropiada como unidad. Emplean técnicas de prueba que recorren caminos específicos en la estructura de control de los componentes (pruebas estructurales) (RODRÍGUEZ, 2014). El método de prueba utilizado para la realización de esta prueba es Caja blanca y la técnica de prueba contenida en este método que se empleó fue la Técnica del camino básico.

Método de prueba: Caja blanca

Estas tienen como labor el seguimiento de las funciones internas del elemento a analizar. Son de gran importancia para el chequeo del estado de la plataforma y verifica la correcta implementación de las unidades internas, las estructuras y sus relaciones y hacen énfasis en la reducción de errores internos. Estas pruebas se caracterizan principalmente porque son los propios sistemas y aplicaciones quienes exponen sus métricas para que el usuario pueda leerlas, analizarlas y tomar decisiones y acciones en función de la obtención de un resultado u otro.

Técnica de prueba: Camino básico

La técnica del Camino básico permite obtener una medida de la complejidad lógica de la codificación de software y usar esa medida como guía para la definición de un

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

conjunto básico de caminos de ejecución independiente en un componente o programa. Un camino o ruta es una vía por la cual procede la ejecución a través de una función desde su inicio hasta el fin (PRESSMAN, 2002).

```
60 def installDHCP(host,user,password,comando): //I
61
62     client = paramiko.SSHClient() //1
63     client.set_missing_host_key_policy(paramiko.AutoAddPolicy()) //2
64     client.connect(host, username=user, password=password) //3
65
66     stdin, stdout, stderr = client.exec_command(comando, get_pty=True) //4
67
68     stdin.write('ijj\n') //5
69     stdin.flush() //6
70
71     stdin.write('S\n') //7
72     stdin.flush() //8
73
74     result = '' //9
75
76
77     for line in stdout.read().splitlines(): //10
78         result += f'{line} ' //11
79         print(result) //12
80
81     e = stderr.read().decode() //13
82     data = {} //14
83     'salida': result, //15
84     'error': e //15
85 //16
86     print(data) //17
87
88     return data //F
89
```

Figura 11: Código de implementación para el método del camino básico. Elaboración propia.

A continuación, se describen los pasos que se realizaron para desarrollar la técnica del camino básico:

1. **Confeccionar el grafo de flujo:** usando el código de la figura 11 se realizó la representación del grafo de flujo, el cual está compuesto por los siguientes elementos:

Nodos: son círculos que representan una o más sentencias procedimentales.

Aristas: son flechas que representan el flujo de control y son análogas a las flechas del diagrama de flujo.

Regiones: son las áreas delimitadas por aristas y nodos.

A continuación, la Figura 11 muestra el grafo de flujo obtenido:

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

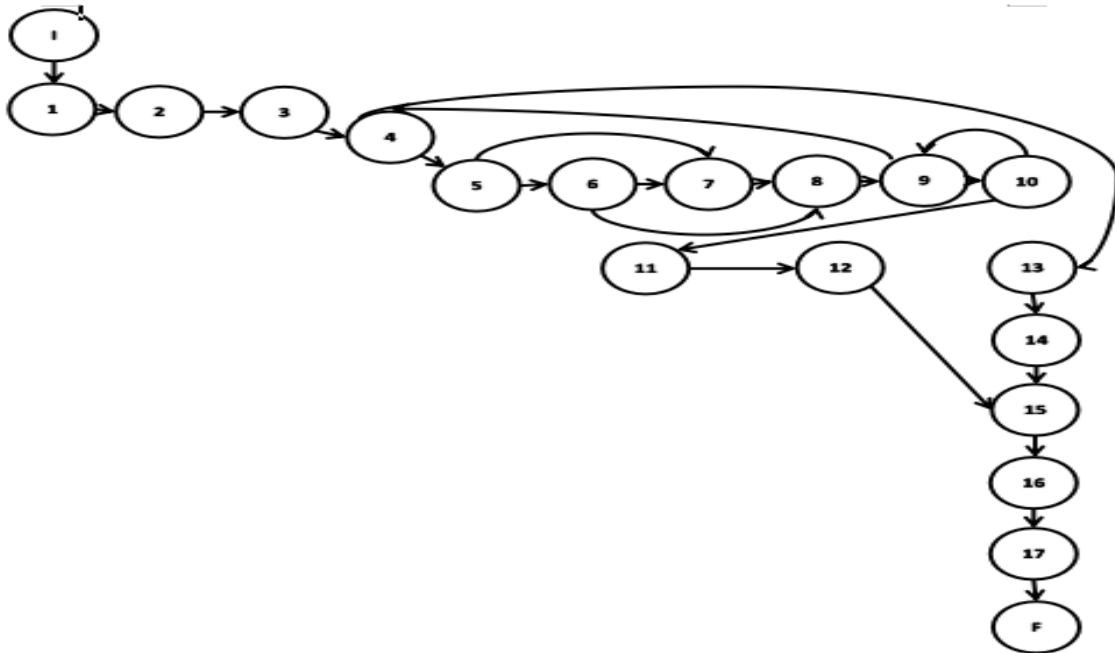


Figura 12: Grafo resultante del método caja blanca camino básico. Elaboración propia.

1. Calcular la complejidad ciclomática

Es una métrica del software en ingeniería del software que proporciona una medición cuantitativa de la complejidad lógica de un programa. El valor calculado define el número de caminos independientes del conjunto básico de un programa, la complejidad ciclomática se calcula de tres formas distintas. En la solución propuesta se aplican las tres variantes con el propósito de triangular los resultados obtenidos, validando que la cantidad de caminos a definir es la correcta:

Variante 1:

$V(G) = A - N + 2$, donde A es el número de aristas del grafo de flujo y N es el número de nodos del mismo $V(G) = A - N + 2 = 23 - 17 + 2 = 8$

Variante 2:

Teniendo en cuenta el número de regiones que presenta el grafo de flujo obtenido, se definen un total de 8 regiones.

Variante 3:

$V(G) = \text{Nodos de predicado} + 1$, donde los nodos predicados son los que tienen como salida más de una arista. $V(G) = \text{Nodos de predicado} + 1$ $V(G) = 7 + 1 = 8$.

Las tres variantes aplicadas brindan como resultado que el valor de la complejidad ciclomática de la función `installDHCP (host, user, password, comando)` es igual a 8. Por lo tanto, se evalúa desde el punto del riesgo de un programa simple, sin mucho riesgo, según la siguiente Tabla que predefinida por McCabe según rangos de evaluación.

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

Complejidad Ciclomática	Evaluación del Riesgo
1-10	Programa simple, sin mucho riesgo
11-20	Más complejo, riesgo moderado
21-50	Complejo, programa de alto riesgo
50	Programa no testeable, muy alto riesgo

1. **Determinar un conjunto básico de caminos linealmente independientes:** una vez aplicadas las tres variantes para calcular la complejidad ciclomática del método installDHCP (host, user, password, comando), se obtiene que el número de caminos de la estructura de control del método, en este caso son 8 caminos, definiéndose de la siguiente forma:

Camino básico #1: I-1-2-3-4-5-6-8-9-4-5-6-7-8-9-10-11-12-15-16-17-F

Camino básico #2: I-1-2-3-4-5-6-7-8-9-10-9-4-13-14-15-16-17-F

Camino básico #3: I-1-2-3-4-5-6-7-8-9-10-11-12-15-16-17-F

Camino básico #4: I-1-2-3-4-5-6-7-8-9-10-11-12-15-16-17-F

Camino básico #5: I-1-2-3-4-5-7-8-9-10-11-12-15-16-17-F

Camino básico #6: I-1-2-3-4-5-6-8-9-4-13-14-15-16-17-F

Camino básico #7: I-1-2-3-4-5-6-8-9-10-11-12-15-16-17-F

Camino básico #8: I-1-2-3-4-5-6-8-9-4-5-6-7-8-9-10F

Es importante aclarar que, aunque se determine como cota máxima 8 caminos no quiere decir que todos sean independientes, por tanto, solo se debe garantizar que en cada camino se añada una nueva arista. Por tanto, se pueden reducir a los siguientes caminos únicos:

Camino básico #1: I-1-2-3-4-5-6-7-8-9-10-11-12-15-16-17-F

Camino básico #2: I-1-2-3-4-13-14-15-16-17-F

Camino básico #3: I-1-2-3-4-5-7-8-9-10-11-12-15-16-17-F

Camino básico #4: I-1-2-3-4-5-6-8-9-10-11-12-15-16-17-F

Camino básico #5: I-1-2-3-4-5-6-7-8-9-4-13-14-15-16-17-F

Camino básico #6: I-1-2-3-4-5-6-7-8-9-10-9-10-11-12-15-16-17-F

Por tanto, existen solo 6 caminos de forma independientes

2. **Determinar un conjunto básico de caminos linealmente independientes:**

Obtención de casos de prueba: cada camino independiente es un caso de prueba (CP) a realizar, de forma que los datos introducidos provoquen que se visiten las sentencias

Capítulo 2: Análisis y diseño del módulo para la administración remota del servicio DHCP en la plataforma Nova-ARST

vinculadas a cada nodo del camino. En este caso se obtuvo 6 caminos independientes, por tanto, se hace necesario la confección de igual número de CP, para aplicar las pruebas a esta función. A continuación, en la Tabla 5 se muestra el CP diseñado para los 6 caminos básicos linealmente independientes obtenido.

3. Tabla 6: Resultados de las pruebas por casos de prueba. Elaboración propia

Capítulo 3. Implementación y pruebas al módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST

No.	Descripción	Condición de ejecución	Entradas				Resultado esperado	Resultado real
			host	user	password	comando		
1	No se introduce en el código el valor de la variable host	El administrador da clic en el botón instalar	" "	"rfrodri guez"	"server@ 1"	"sudo apt- get install isc-dhcp- server"	Alerta de error	Correcto
2	No se introduce en el código el valor de la variable usuario		"196.1 0.102. 24 "	" "	"server@ 1 "	"sudo apt- get install isc-dhcp- server"	Alerta de error	Correcto
3	No se introduce en el código el valor de la variable contraseña		"196.1 0.102. 24 "	"mdela o "	" "	"sudo apt- get install isc-dhcp- server"	Alerta de error	Correcto
4	No se introduce en el código el valor de la variable comando		"196.1 0.102. 24 "	"mdela o "	"server@ 1 "	" "	Alerta de error	Correcto
5	Se introducen todos los valores de la variables		"196.1 0.102. 24 "	"rfrodri guez"	"server@ 1"	"sudo apt- get install isc-dhcp- server"	Se instala el servicio DHCP	Correcto
6	No se introduce ningún valor de variable		" "	" "	" "	" "	Alerta de error	Correcto

Capítulo 3. Implementación y pruebas al módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST

3.5.2 Pruebas funcionales o de Caja negra

Las pruebas funcionales son aquellas que se llevan a cabo sobre la interfaz del software sin prestar atención al código, por lo que los casos de prueba son creados con el objetivo de demostrar que la entrada es aceptada de forma adecuada y que se produce una salida correcta. El diseño de esta prueba se realiza con la intención de detectar funciones incorrectas o ausentes, errores en accesos a bases de datos externas, errores de interfaz, errores de rendimiento, y errores de inicialización y de terminación (PRESSMAN, 2002).

El método de prueba utilizado para la realización de esta prueba es Caja negra y la técnica de prueba contenida en este método que se empleó fue la de Partición de equivalencia.

Técnica de prueba: Partición de equivalencia

Permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos. La técnica divide el dominio de entrada de un programa en clases de datos, a partir de las cuales pueden derivarse casos de prueba. Además, descubre clases de errores, que, de otra manera, requeriría la ejecución de muchos casos antes de que se observe el error general. Mediante su empleo se puede reducir al máximo el total de casos de prueba que deben desarrollarse.

Diseños de casos de pruebas

El diseño de casos de prueba, tiene un único objetivo: tener la mayor probabilidad de encontrar el mayor número de errores con la mínima cantidad de esfuerzo y tiempo posible. Se busca crear un conjunto de casos de prueba que sean efectivos descubriendo defectos en los programas y muestren que el sistema satisface sus requerimientos. Para su diseño, se selecciona una característica del sistema o componente que se está probando, un conjunto de entradas que ejecutan dicha característica, se documentan las salidas esperadas o rasgos de salida y donde sea posible se diseña una prueba automatizada que demuestre que las salidas reales y las esperadas son las mismas. A continuación, se presenta el diseño de caso de prueba para el RF 1. Instalar servicio DHCP, Tabla 6.

Tabla 7: Caso de prueba Funcional del RF 1 Instalar servicio DHCP. Elaboración propia.

Escenario (Front end)	Descripción	Variable host	Variable usuario	Variable contras eña	Variable comando	Respuesta del sistema
--------------------------------------	--------------------	--------------------------	-----------------------------	-------------------------------------	-----------------------------	----------------------------------

Capítulo 3. Implementación y pruebas al módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST

EC 1.1	La plataforma debe mostrar la vista de: Bienvenido al módulo DHCP en caso de que no se haya instalado en el servidor	V 192.168. 137.158	V Nova	V 1234567 890	V sudo apt-get install-dhcp-server	El servicio DHCP es instalado de forma correcta en el servidor
EC 1.2	La plataforma muestra la opción de instalar nuevamente el servicio DHCP	192.168. 137.158	Nova	1234567 890	sudo apt-get install-dhcp-server	El servicio DHCP es instalado de forma correcta en el servidor

Con el objetivo de comprobar que las funcionalidades del sistema fueron implementadas correctamente y responden a las necesidades del cliente, aplicando los diseños de casos de prueba antes descritos, se realizaron pruebas funcionales. Las pruebas se realizaron en dos iteraciones. En la primera se detectaron un total de 20 No Conformidades (NC), clasificadas en 4 de ortografía, 10 de redacción y 3 de validación, al finalizar la iteración todas las NC quedaron resueltas. En la segunda iteración los resultados fueron satisfactorios, obteniéndose cero NC. Figura 13 ilustra los resultados de aplicar el método de caja negra, teniendo en cuenta los tipos de NC identificadas (ortografía, redacción y validación).

Capítulo 3. Implementación y pruebas al módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST

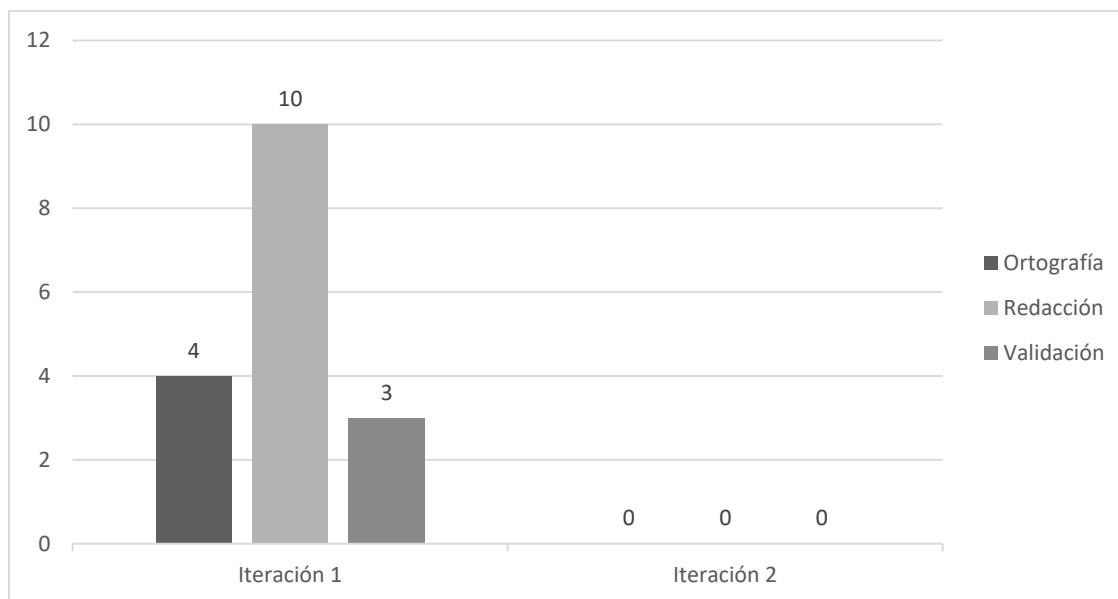


Figura 13: No Conformidades detectadas al aplicar el método de caja negra. Elaboración propia.

3.5.3 Pruebas de aceptación

En el desarrollo ágil, las pruebas de aceptación forman parte del proceso y no son una ocurrencia tardía. Sin embargo, la intención sigue siendo la misma: verificar que el software cumple las expectativas desde el punto de vista del cliente y de los usuarios finales. Esta es la etapa final en el proceso de pruebas, antes de que el sistema sea aceptado para uso operacional. El sistema se pone a prueba con datos suministrados por el cliente del sistema, en vez de datos de prueba simulados. Las pruebas de aceptación revelan los errores y las omisiones en la definición de requerimientos del sistema, ya que los datos reales ejercitan el sistema en diferentes formas a partir de los datos de prueba (SOMMERVILLE, 2007).

Tabla 8: Caso de Prueba de aceptación para la historia de usuario "Instalar servicio DHCP".
Elaboración propia

Caso de prueba de aceptación
Nombre de la historia de usuario: Instalar servicio DHCP
Nombre de la persona que realiza la prueba: Lexys Manuel Díaz Alonso
Descripción de la prueba: La plataforma permite instalar el servicio DHCP
Condiciones de ejecución: El servicio DHCP no puede haber sido instalado en la PC a la cual se desea emplear este servicio.
Entrada/Pasos de ejecución:

Capítulo 3. Implementación y pruebas al módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST

1- El administrador entra al módulo DHCP
2- El administrador pincha el botón de Instalar ahora
Resultado esperado: La plataforma ejecuta internamente el comando en consola que permite instalar el servicio DHCP
Evaluación de la prueba: Satisfactoria

Al terminar las pruebas de aceptación se obtuvo un total de 8 No Conformidades en la primera iteración, de las cuales 3 fueron resueltas, 3 no procedían y 2 pendiente. En la segunda iteración se obtuvieron 5 no conformidades donde 3 fueron resueltas, 1 no procedía y 1 pendiente, en la tercera iteración se encontraron 3 no conformidades y en una cuarta iteración fueron todas resueltas. Durante las 4 iteraciones no quedó ninguna no conformidad por resolver. Los resultados correspondientes a estas pruebas se muestran en la Figura 14.

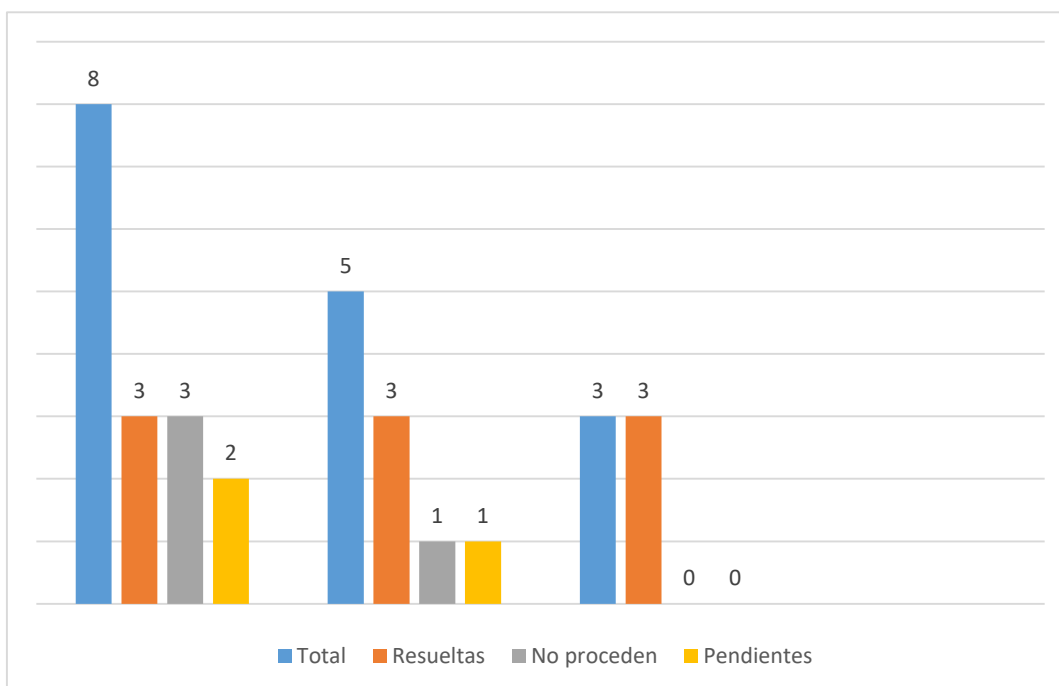


Figura 14: Resultados de las pruebas de aceptación. Elaboración propia.

3.6 Evaluación del objetivo general de la investigación

El cuestionario empleado para determinar el grado de satisfacción de la plataforma Nova-ARST cuenta con un total de cinco preguntas, de ella tres cerradas (1, 3 y 5) y dos abiertas (2 y 4), cuya relación ignora el sujeto. Estas tres preguntas cerradas se relacionan a través del “Cuadro lógico de ladov” que se muestra en la Tabla 10. El número resultante de la interrelación de las tres preguntas nos indica la posición de cada sujeto en la escala de satisfacción, o sea su satisfacción individual. La aplicación de esta

Capítulo 3. Implementación y pruebas al módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST

técnica constituye una vía indirecta para el estudio de satisfacción, ya que los criterios que se utilizan se fundamentan en las relaciones que se establecen entre las tres preguntas cerradas, que se intercalan dentro de un cuestionario que se muestra en el Anexo 4 y cuya relación el encuestado desconoce.

Antes de comenzar el desarrollo e implementación de la plataforma se le preguntó al cliente sobre las expectativas que tenían con la misma, las respuestas fueron las siguientes:

- ✓ Nos va ayudar en el desarrollo de la informática en la empresa, es una plataforma que organizará nuestra red.
- ✓ Aprender temas nuevos, muy útiles para el desarrollo de nuestra profesión como administradores de redes y seguridad informática.
- ✓ Pienso que nos será muy útil para la organización de nuestra red empresarial.
- ✓ Conocer y dominar sitios WEB propios de la especialidad.
- ✓ Desarrollar habilidades en cuanto el uso del SO Nova Servidor 8.0.
- ✓ Obtener métodos y el conocimiento necesario para hacer un uso más eficiente de las nuevas tecnologías.
- ✓ Es una plataforma básica para aprender a buscar información de nuestra red empresarial interna, con nuevas herramientas y recursos y hacer más fácil el trabajo y el conocimiento de servicios telemáticos.

La encuesta elaborada para evaluar el índice de satisfacción de los usuarios potenciales de la propuesta de solución fue aplicada a 7 especialistas del proyecto Nova. La escala de satisfacción utilizada es la siguiente:

1. Clara satisfacción, 2. Más satisfecho que insatisfecho, 3. No definida, 4. Más insatisfecho que satisfecho, 5. Clara insatisfacción, 6. Contradictoria

Tabla 9: Cuadro lógico de IADOV. Elaboración propia.

1- ¿Luego de haber mostrado los resultados de la solución, en qué medida gusta la solución desarrollada?	2- ¿Considera usted correcta la forma en que se realiza el módulo del servicio telemático DHCP en la plataforma Nova-ARST?		
	No	No sé	Sí

Capítulo 3. Implementación y pruebas al módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST

	3- ¿Considera usted factible la implementación de un módulo que permita la administración remota del servicio telemático DHCP en la plataforma Nova-ARST?								
	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Clara satisfacción	1	2	6	2	6	6	6	6	6
Más satisfecho que insatisfecho	2	2	3	2	3	3	6	3	3
No definida	3	3	3	3	3	3	3	3	3
Más insatisfecho que satisfecho	6	3	6	3	4	4	3	3	4
Clara insatisfacción	6	6	6	6	4	4	6	4	5
Contradictoria	2	3	6	3	3	3	6	3	4

La forma de utilizar la tabla es la siguiente, cada encuestado recibe una evaluación individual en dependencia de las respuestas que dé a las preguntas cerradas. Para facilitar el procesamiento posterior, en el diseño de la encuesta se debe tener en cuenta que a estas preguntas solo se responda de la forma prevista en el cuadro lógico de IADOV. Las respuestas a las preguntas 2 y 3 pueden ser Sí, No, No sé, y a las preguntas 4, “Clara satisfacción”, “Más satisfecho que insatisfecho”, “No definida”, “Más insatisfecho que satisfecho”, “Clara insatisfacción”, o “Contradictoria”.

Para obtener el índice de satisfacción grupal (ISG) se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y - 1. El número resultante de la interrelación de las tres preguntas que indica la posición de cada encuestado en la siguiente escala de satisfacción:

- ✓ Máxima satisfacción +1
- ✓ Más satisfecho que insatisfecho +0.5
- ✓ No definido y contradictorio 0
- ✓ Más insatisfecho que satisfecho -0.5
- ✓ Máxima insatisfacción -1

La satisfacción grupal se calcula por la siguiente ecuación:

$$ISG = (A (+1) + B (+0.5) + C (0) + D (-0.5) + E (-1)) / N$$

Capítulo 3. Implementación y pruebas al módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST

En esta fórmula A, B, C, D, E, representan la cantidad de encuestados colocados respectivamente en las posiciones de satisfacción 1; 2; 3 ó 6; 4; 5 y donde N representa la cantidad total de encuestados. El índice grupal arroja valores entre + 1 y - 1. Los valores que se encuentran comprendidos entre - 1 y - 0,5 indican insatisfacción; los comprendidos entre - 0,49 y + 0,49 evidencian contradicción y los que caen entre 0,5 y 1 indican que existe satisfacción.

Resultados obtenidos

Los resultados obtenidos de la aplicación de la encuesta se presentan en la Tabla 11.

Tabla 11: Resultados obtenidos de los encuestados. Elaboración propia.

Categorías grupales de satisfacción	N = 7	Escala
Máxima satisfacción	4	A
Más satisfecho que insatisfecho	2	B
No definido y contradictorio	1	C
Más insatisfecho que satisfecho	0	D
Máxima insatisfacción	0	E

Cálculo del ISG

$$\text{ISG} = A (+1) + B (+0.5) + C (1) / N$$

$$\text{ISG} = (4(+1) + 2(+0.5)) + 1(1) / 7 = 0.57$$

Interpretación del resultado del ISG

El proceso de evaluación del objetivo de la investigación mediante la técnica de IADOV confirmó su factibilidad de uso, expresando cuantitativamente que existe satisfacción en el ISG (0.57) y cualitativamente en los criterios emitidos en el centro de desarrollo CESOL, lo que refleja la aceptación de la propuesta, como se muestra en el Anexo 5 y el reconocimiento a su utilidad.

Capítulo 3. Implementación y pruebas al módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST

3.7 Interfaz principal

Una vez finalizado el desarrollo del software es posible visualizar la pantalla principal del módulo del servicio telemático DHCP para la administración remota en la plataforma Nova-ARST, donde se observa el resultado obtenido durante la implementación de las historias de usuario descritas en el capítulo anterior.

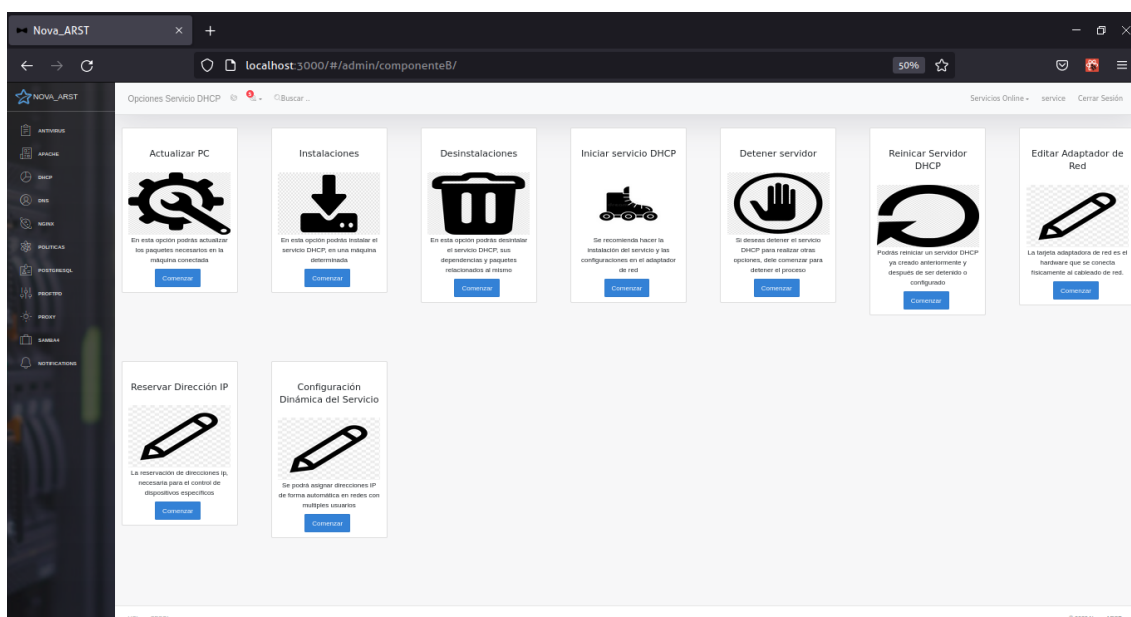


Figura 15: Interfaz principal del módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST. Elaboración propia

3.8 Conclusiones del capítulo

En este capítulo se abordaron los elementos de la implementación del módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST, así como las pruebas realizadas al mismo y los resultados obtenidos, por lo que se puede concluir:

- ✓ La elaboración del diagrama de componentes permitió una mejor comprensión de la estructura de los componentes de la propuesta de solución.
- ✓ La implementación del módulo facilitó la obtención de una aplicación correctamente funcional para su uso.
- ✓ La aplicación de las pruebas unitarias, integración, funcionales, seguridad y aceptación permitieron detectar errores que afectaban el funcionamiento del

Capítulo 3. Implementación y pruebas al módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST

módulo, lo que permitió corregirlos a tiempo para que el mismo cumpliera con los requisitos funcionales definidos en la etapa de análisis.

Conclusiones

De manera general se puede concluir sobre la presente investigación:

- ✓ El análisis de los referentes teóricos y de las herramientas informáticas que hacen uso del servicio telemático DHCP estudiadas en otras plataformas web evidenció la necesidad de desarrollar un módulo que permitiera la administración remota a la plataforma Nova-ARST.
- ✓ La selección de herramientas, lenguajes y tecnologías permitió la implementación del módulo del servicio telemático DHCP para la administración remota en la plataforma Nova-ARST.
- ✓ Las pruebas diseñadas y ejecutadas permitieron detectar las deficiencias presentes en el módulo desarrollado, corrigiéndose las mismas logrando un producto más seguro y funcional.
- ✓ La aplicación de la técnica de IADOV propició la evaluación satisfactoria del módulo del servicio telemático DHCP para la administración remota en la plataforma Nova-ARST.
- ✓ El desarrollo del módulo DHCP redujo los pasos de configuración inicial y recurrente de este servicio telemático.
- ✓ La manera visual en que se muestra el servicio telemático permitió la reducción de tiempo de uso en administradores de red con poca experiencia.
- ✓ La validación de datos como muestra de ayuda por parte del módulo desarrollado, garantizó la no introducción de comandos o datos erróneos.

Recomendaciones

Una vez concluido el trabajo de diploma se recomienda lo siguiente:

- ✓ Profundizar en el estudio del uso de comandos por consola para determinar otros requisitos que pudieran enriquecer aún más el módulo del servicio telemático DHCP, para que tenga un amplio abanico de opciones que hagan más robusta su administración remota a través del protocolo SSH en la plataforma Nova-ARST.

Capítulo 3. Implementación y pruebas al módulo para la administración remota del servicio telemático DHCP en la plataforma Nova-ARST

Referencias bibliográficas

- Chakray.** Lenguajes de programación: tipos y características. [En línea] 2015. [Consultado el: 10 de mayo de 2022]. Disponible en: <https://www.chakray.com/es/lenguajes-programacion-tipos-caracteristicas/>
- Delftstack** Python Paramiko. [En línea] 2020 [Consultado el: 21 de mayo de 2022] Disponible en: <https://www.delftstack.com/es/howto/python-pandas/paramiko-python/>
- Deyimar**, A. Hostinger. ¿Cómo funciona el SSH? [En línea] 2022. [Consultado el: 7 de mayo de 2022]. Disponible en: <https://www.hostinger.es/tutoriales/que-es-ssh>.
- Django.** The web framework for perfectionists with deadlines. [En línea] 2022. [Consultado el: 02 de junio de 2022.] <https://www.djangoproject.com/start/overview/>
- Django Rest Framework.** [En línea]. 2022. [Consultado el: 21 de mayo de 2022]. Disponible en: <https://www.django-rest-framework.org/>
- Garro**, Arkaitz. Open Libra. HTML 5. [En línea] 2015. [Consultado el: 10 de mayo de 2022.] Disponible en: <https://openlibra.com/es/book/html5>
- Gómez**, Ramiro A. Lenguajes de programación. [En línea] 2017. [Consultado el: 02 de mayo de 2022.] Disponible en: <http://www.ilustrados.com/documentos/lenguajes-programacion-020608.pdf>.
- Lucidchart.** Qué es el lenguaje unificado de modelado (UML) _ Lucidchart. [En línea]. 2017. [Consultado el: 7 de mayo de 2022]. Disponible en: <https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml>.
- Maida** EG, Pacienza, J. Metodologías de desarrollo de software [en línea]. Tesis de Licenciatura en Sistemas y computación. Facultad de Química e Ingeniería Fray Rogelio Bacon. Universidad Católica de Argentina, 2015. [Consultado el: 16 de mayo de 2022] Disponible en: <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>
- Mdn.** Tecnologías para desarrolladores web. [En línea] 2022. [Consultado el: 10 de mayo de 2022]. Disponible en: <https://developer.mozilla.org/es/docs/Web/JavaScript>
- Navajas Ojeda, Antonio. Guía completa de CSS. [En línea] 2012. . [Consultado el: 02 de mayo de 2022.] Disponible en: <https://openlibra.com/es/book/guia-completa-de-css3>
- Pressman**, Roger. Ingeniería de Software: Un Enfoque Práctico. Quinta Edición. Nueva York, Estados Unidos: McGraw-Hill, 2002, p. 207-384.
- React.** Una biblioteca de JavaScript para construir interfaces de usuario. [En línea]. 2022. [Consultado el 10 de mayo de 2022] Disponible en: <https://es.reactjs.org/>
- Rodríguez**, Tamara. Metodología de desarrollo para la actividad productiva en la UCI. La Lisa, La Habana, Cuba, 2014

Referencias Bibliográficas

Teorema, Análisis y diseño. [En línea] 2020. [Consultado el: 1 de septiembre de 2022.] Disponible en: <https://teorema-rd.com/producto/curso-analisis-y-diseno-de-sistemas>

Lucidchart, Diseño arquitectónico. [En línea] 2022. [Consultado el: 1 de septiembre de 2022.] Disponible en: <https://www.lucidchart.com/blog/es/como-disenar-una-arquitectura-de-software>

LÓPEZ, La dirección estratégica de la empresa. Teoría y aplicaciones. [En línea] 2015. [Consultado el: 3 de septiembre de 2022.] Disponible en: http://www.guerrasynavas.es/pdf/cede_recension.pdf

Francisco José, García Peñalvo **Carlos**, Pardo Aguilar. Diagramas de Clase en UML 1.1. [En línea] 2022. [Consultado el: 4 de septiembre de 2022.] Disponible en: <https://repositorio.grial.eu/bitstream/grial/353/1/DClase.pdf>

LARMAN, Craig. UML y patrones: Una introducción al análisis y diseño orientado a objetos y al proceso unificado. Segunda edición. 1999. México. Prentice Hall. ISBN 970-17-0261-1. p.164-238

CinergixPty.Ltd., Blog Creately-Diagramas. [En línea] 2008-2022, [Consultado el: 28 de octubre de 2022.] Disponible en: <https://creately.com/blog/es/diagramas/tutorial-de-diagrama-de-componentes/>

Programador clic, Estándares de codificación. [En línea] 2020-2022, [Consultado el: 30 de octubre de 2022.] Disponible en: <https://programmerclick.com/article/>


Agestic, versión 1.7, ¿Cómo evaluar y comparar software? Editorial Presidencia, República Oriental del Uruguay. [En línea] 2019, [Consultado el: 1 de noviembre de 2022.] Disponible en: <https://www.gub.uy/agencia-gobierno-electronico-sociedad-informacion-conocimiento/sites/agencia-gobierno-electronico-sociedad-informacionconocimiento/files/documentos/publicaciones/Gu%C3%ADa%20para%20la%20evaluaci%C3%B3n%20y%20comparaci%C3%B3n%20de%20software%201.7.pdf>

Anexos

Anexo 1: Entrevista realizada al especialista encargado del despliegue de la plataforma Nova-ARST


1. ¿Por qué usar un SO que no sea de pago y desarrollado por Cuba?
2. ¿Por qué crear una plataforma que garantice la administración remota de los servidores DHCP en las diferentes empresas que hagan uso de este SO?
3. ¿Afecta la existencia de varios servidores DHCP en una misma empresa a la hora de administrar una red?
4. ¿Qué ventajas traería una plataforma que asigne de forma automática y no manual las direcciones IP?
5. ¿Existen algunas otras plataformas que tengan funcionalidades similares a la que se quiere lograr?
6. ¿Cómo se llamaría esta nueva plataforma?
7. ¿Ayudaría económicamente al país una plataforma de estas características?

Anexo 2: Historia de usuario. Desinstalar servicio DHCP.

Historia de usuario	
Numero: HU_2	Nombre: Desinstalar servicio DHCP
Prioridad: Alta	Iteración Asignada: 1
Programador: Reynier Fernández Rodríguez	Tiempo Estimado: 1 semana
Riesgo en Desarrollo: N/A	Tiempo Real: 1 semana
Descripción: El sistema comienza el proceso de desinstalación del servicio DHCP una vez el administrador pincha esta opción	
Observaciones: El administrador debe acceder al módulo DHCP de la plataforma Nova-ARST y en caso de querer desinstalar el servicio DHCP pinchar el botón Comenzar ✓ Verificar que la máquina en la que se encuentra tenga conexión a <i>internet</i> ✓ Confirmar desinstalación	
Prototipo: <div style="border: 1px solid gray; padding: 10px; text-align: center;">  </div>	

Anexo 3: Historia de usuario. Desinstalar servicio DHCP.

Historia de usuario	
Numero: HU_2	Nombre: Desinstalar servicio DHCP

Prioridad: Alta	Iteración Asignada: 1
Programador: Reynier Fernández Rodríguez	Tiempo Estimado: 1 semana
Riesgo en Desarrollo: N/A	Tiempo Real: 1 semana
Descripción: El sistema comienza el proceso de desinstalación del servicio DHCP una vez el administrador pincha esta opción	
Observaciones: El administrador debe acceder al módulo DHCP de la plataforma Nova-ARST y en caso de querer desinstalar el servicio DHCP pinchar el botón Comenzar	
<ul style="list-style-type: none"> ✓ Verificar que la máquina en la que se encuentra tenga conexión a <i>internet</i> ✓ Confirmar desinstalación 	
Prototipo:	
	

Anexo 4: Historia de usuario. Configurar parámetros del servicio DHCP.

Historia de usuario	
Numero: HU_3	Nombre: Configurar parámetros del servidor DHCP
Prioridad: Alta	Iteración Asignada: 1
Programador: Reynier Fernández Rodríguez	Tiempo Estimado: 1 semana
Riesgo en Desarrollo: N/A	Tiempo Real: 1 semana
Descripción: La plataforma a través de la opción de configuración de parámetros del servidor DHCP permite modificar las propiedades del servidor creado	
Observaciones: El administrador debe acceder al módulo DHCP de la plataforma Nova-ARST y en caso de querer configurar los parámetros del servicio DHCP pinchar el botón Comenzar	
<ul style="list-style-type: none"> ✓ Verificar que la máquina en la que se encuentra tenga conexión a <i>internet</i> ✓ Comenzar ✓ Entrada de datos en el campo red, subred, puerta de enlace, máscara de red, máscara de subred, nombre de dominio, rango de direcciones IP, tiempo de arrendamiento del servicio. ✓ Confirmar con el botón guardar 	
Prototipo:	

Anexo 5: Historia de usuario. Mostrar lista de arrendamientos activos

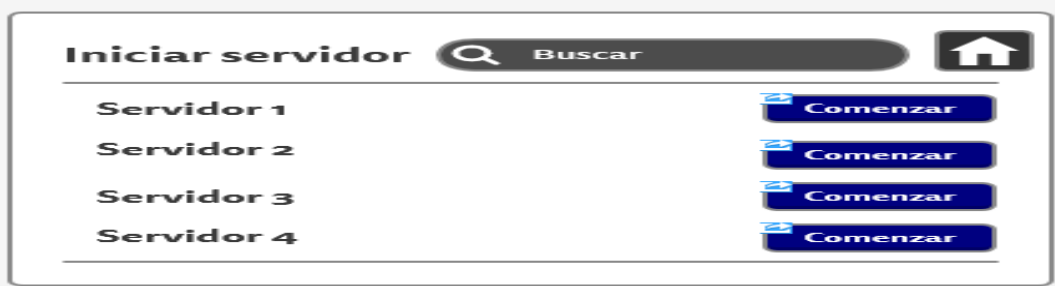
Historia de usuario	
Numero: HU_4	Nombre: Mostrar lista de arrendamientos activos
Prioridad: Alta	Iteración Asignada: 1
Programador: Reynier Fernández Rodríguez	Tiempo Estimado: 1 semana
Riesgo en Desarrollo: N/A	Tiempo Real: 1 semana
Descripción: La plataforma a través de esta opción muestra al administrador todas las máquinas que están usando el servidor DHCP, así como el tiempo disponible para usar el servidor DHCP.	
Observaciones: El administrador debe acceder al módulo DHCP de la plataforma Nova-ARST y en caso de querer mostrar las máquinas que están usando el servidor DHCP pinchar el botón Acceder	
<ul style="list-style-type: none"> ✓ Verificar que la máquina en la que se encuentra tenga conexión a <i>internet</i> ✓ Acceder 	
Prototipo:	

Anexo 6: Historia de usuario. Arrancar servidor DHCP.


Historia de usuario	
Numero: HU_5	Nombre: Arrancar servidor DHCP
Prioridad: Alta	Iteración Asignada: 1
Programador: Reynier Fernández Rodríguez	Tiempo Estimado: 1 semana
Riesgo en Desarrollo: N/A	Tiempo Real: 1 semana
Descripción: La plataforma a través de esta opción permite al administrador arrancar los servidores DHCP ya creados.	
Observaciones: El administrador debe acceder al módulo DHCP de la plataforma Nova-ARST y en caso de querer arrancar el servidor DHCP pinchar el botón comenzar	

- ✓ Verificar que la máquina en la que se encuentra tenga conexión a *internet*
- ✓ Comenzar

Prototipo:



Anexo 7: Historia de usuario. Detener servidor DHCP.

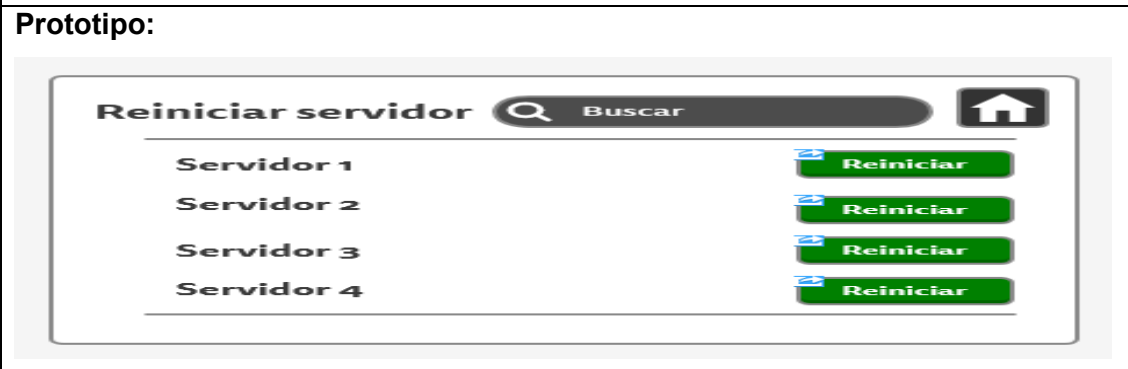
Historia de usuario	
Numero: HU_6	Nombre: Detener servidor DHCP
Prioridad: Alta	Iteración Asignada: 1
Programador: Reynier Fernández Rodríguez	Tiempo Estimado: 1 semana
Riesgo en Desarrollo: N/A	Tiempo Real: 1 semana
<p>Descripción: La plataforma a través de esta opción permite al administrador detener los servidores DHCP ya creados.</p>	
<p>Observaciones: El administrador debe acceder al módulo DHCP de la plataforma Nova-ARST y en caso de querer arrancar el servidor DHCP pinchar el botón detener.</p>	
<ul style="list-style-type: none"> ✓ Verificar que la máquina en la que se encuentra tenga conexión a <i>internet</i> ✓ Detener 	
<p>Prototipo:</p> 	

Anexo 8: Historia de usuario. Reiniciar servidor DHCP.

Historia de usuario	
Numero: HU_7	Nombre: Reiniciar servidor DHCP
Prioridad: Alta	Iteración Asignada: 1
Programador: Reynier Fernández Rodríguez	Tiempo Estimado: 1 semana
Riesgo en Desarrollo: N/A	Tiempo Real: 1 semana
<p>Descripción: La plataforma a través de esta opción permite al administrador reiniciar los servidores DHCP ya creados.</p>	

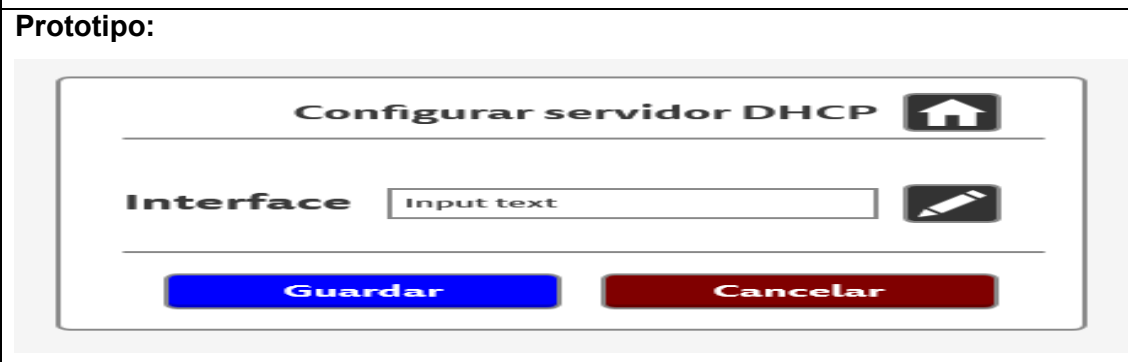
Observaciones: El administrador debe acceder al módulo DHCP de la plataforma Nova-ARST y en caso de querer reiniciar el servidor DHCP pinchar el botón detener.

- ✓ Verificar que la máquina en la que se encuentra tenga conexión a *internet*
- ✓ Detener




Anexo 9: Historia de usuario. Reiniciar servidor DHCP.

Historia de usuario	
Numero: HU_9	Nombre: Configurar interfaz internet
Prioridad: Alta	Iteración Asignada: 1
Programador: Reynier Fernández Rodríguez	Tiempo Estimado: 1 semana
Riesgo en Desarrollo: N/A	Tiempo Real: 1 semana
Descripción: La plataforma a través de esta opción permite al administrador modificar el adaptador de red IPv4.	
Observaciones: El administrador debe acceder al módulo DHCP de la plataforma Nova-ARST y en caso de querer modificar el adaptador de internet del servidor DHCP pinchar el botón comenzar.	
<ul style="list-style-type: none"> ✓ Verificar que la máquina en la que se encuentra tenga conexión a <i>internet</i> ✓ Introducir nuevo adaptador de internet ✓ Guardar 	



Anexo 10: Historia de usuario. Añadir máquinas al servidor DHCP.

Historia de usuario	
Numero: HU_10	Nombre: Añadir maquinas
Prioridad: Alta	Iteración Asignada: 1
Programador: Reynier Fernández Rodríguez	Tiempo Estimado: 1 semana

Riesgo en Desarrollo: N/A	Tiempo Real: 1 semana
Descripción: La plataforma a través de esta opción permite al administrador reservar una dirección IP de forma fija, suele usarse en impresoras, servidores, etc.	
Observaciones: El administrador debe acceder al módulo DHCP de la plataforma Nova-ARST y en caso de querer reservar una dirección IP para un dispositivo determinado en el servidor DHCP pinchar el botón comenzar.	
<ul style="list-style-type: none"> ✓ Verificar que la máquina en la que se encuentra tenga conexión a <i>internet</i> ✓ Introducir Mac del cliente ✓ La dirección IP fija ✓ Guardar 	
Prototipo:	
	

Anexo 11: Estandar de codificación. Propiedades. Iteración en línea.

```

1  return (
2    <div>
3      {this.props.data.map(function(data, i) {
4        return (<Component data={data} key={i} />)
5      })}
6    </div>
7  );

```

Anexo 9: Estandar de codificación. Comentarios.

```

1  var content = (
2    <Nav>
3      {/** child comment, put {} around */}
4      <Person
5        /** multi
6          line
7          comment */
8        name={window.isLoggedIn ? window.name : ''} // end of line comment
9      />
10   </Nav>
11 );
12

```

Anexo 10: Estandar de codificación. Uso de comillas.

```

1 // bad
2 <Foo bar='bar' />
3 // good
4 <Foo bar="bar" />
5 // bad
6 <Foo style={{ left: "20px" }} />
7 // good
8 <Foo style={{ left: '20px' }} />
9 // JavaScript Expression
10 const person = <Person name={window.isLoggedIn ? window.name : ''} />;
11 // HTML/JSX
12 const myDivElement = <div className="foo" />;
13 const app = <Nav color="blue" />;
14 const content = (
15   <Container>
16     {window.isLoggedIn ? <Nav /> : <Login />}
17   </Container>
18 );

```

Anexo 11: Estandar de codificación. HTML condicional.

```

1 {this.state.show && 'This is Shown'}
2 {this.state.on ? 'On' : 'Off'}

```

Anexo 12: Estandar de codificación. HTML condicional.

```

1 var dinosaurHtml = '';if (this.state.showDinosaurs) { dinosaurHtml = ( <section> <Dino Copiar
2 [
3 ] (https://github.com/minwe/style-guide/blob/master/React.js.md#-use)

```

Anexo 13: Estandar de codificación. () Uso.

```

1 // bad
2 return <div><ComponentOne /><ComponentTwo /></div>;
3 // good
4 var multilineJsx = (
5   <header>
6     <Logo />
7     <Nav />
8   </header>
9 );
10 // good
11 return (
12   <div>
13     <ComponentOne />
14     <ComponentTwo />
15   </div>
16 );

```

Anexo 14: Estandar de codificación. () Uso.

```

1 var singleLineJsx = <h1>Simple JSX</h1>;
2 // good, when single line
3 render() {
4   const body = <div>hello</div>;
5   return <MyComponent>{body}</MyComponent>;
6 }

```

Anexo 15: Estandar de codificación. Etiqueta de cierre automático.

```

1 // bad
2 <Logo></Logo>
3 <Logo/>
4 // very bad
5 <Foo />
6 // bad
7 <Foo
8 />
9 // good
10 <Logo />

```

Anexo 16: Estandar de codificación. Organización del código interno del componente.

```

1 // bad
2 React.createClass({
3   _onClickSubmit() {
4     // do stuff
5   }
6   // other stuff
7 });
8 // good
9 React.createClass({
10  onClickSubmit() {
11    // do stuff
12  }
13  // other stuff
14 });

```

Anexo 17: Estandar de codificación. Organización del código interno del componente.

```

1 // El código está organizado en el siguiente orden en el componente React
2 React.createClass({
3   displayName: '',
4   mixins: [],
5   statics: {},
6   propTypes: {},
7   getDefaultProps() {
8     // ...
9   },
10  getInitialState() {
11    // do something
12  },
13  componentWillMount() {
14    // do something
15  },
16  componentDidMount() {
17    // do something: add DOM event listener, etc.
18  },
19  componentWillReceiveProps() {
20  },
21  shouldComponentUpdate() {},
22  componentWillUpdate() {},
23  componentDidUpdate() {},
24  componentWillUnmount() {
25    // do something: remove DOM event listener. etc.
26  },
27  // clickHandlers or eventHandlers like onClickSubmit() or onChangeDescription()
28  handleClick() {
29    // ...
30  },
31  // getter methods for render like getSelectReason() or getFooterContent()
32  // Optional render methods like renderNavigation() or renderProfilePicture()
33  render() {
34    // ...
35  }
36 });

```